

**APLIKASI *DATA MINING* UNTUK MENAMPILKAN INFORMASI
TINGKAT KELULUSAN MAHASISWA
(Studi Kasus di Fakultas MIPA Universitas Diponegoro)**



SKRIPSI

**Disusun Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Komputer**

**Disusun oleh:
Nuqson Masykur Huda
J2F005280**

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS DIPONEGORO
2010**

HALAMAN PENGESAHAN

Judul : Aplikasi *Data Mining* Untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa (Studi Kasus di Fakultas MIPA Universitas Diponegoro)

Nama : Nuqson Masykur Huda

NIM : J2F005280

Telah diujikan pada sidang Tugas Akhir tanggal 1 September 2010 dan dinyatakan lulus pada tanggal 28 September 2010

Semarang, 28 September 2010
Panitia Penguji Tugas Akhir
Ketua,

Drs. Eko Adi Sarwoko, M.Kom
NIP.196511071992031003

Mengetahui,
Ketua Jurusan Matematika

Mengetahui,
Ketua Program Studi Teknik Informatika

Dr. Widowati, S.Si, M.Si
NIP.196902141994032002

Drs. Eko Adi Sarwoko, M.Kom
NIP.196511071992031003

HALAMAN PENGESAHAN

Judul : Aplikasi *Data Mining* Untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa (Studi Kasus di Fakultas MIPA Universitas Diponegoro)

Nama : Nuqson Masykur Huda

NIM : J2F005280

Telah diujikan pada sidang Tugas Akhir tanggal 1 September 2010

Pembimbing I,

Beta Noranita, S.Si, M.Kom
NIP. 197308291998022001

Semarang, 1 September 2010

Pembimbing II

Nurdin Bahtiar, S.Si, M.T
NIP. 197907202003121 001

ABSTRAK

Pertumbuhan yang pesat dari akumulasi data telah menciptakan kondisi kaya akan data tapi minim informasi. *Data mining* merupakan penambangan atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data dalam jumlah besar yang diharapkan dapat mengatasi kondisi tersebut. Dengan memanfaatkan data induk mahasiswa dan data kelulusan mahasiswa, diharapkan dapat menghasilkan informasi tentang tingkat kelulusan dengan data induk mahasiswa melalui teknik *data mining*. Kategori tingkat kelulusan diukur dari lama studi dan IPK. Algoritma yang digunakan adalah algoritma *apriori*, informasi yang ditampilkan berupa nilai *support* dan *confidence* dari masing-masing kategori tingkat kelulusan.

Kata kunci : *data mining*, algoritma *apriori*, tingkat kelulusan, data induk mahasiswa.

ABSTRACT

The rapid growth of the accumulation of data has created a data-rich condition, but minim of information. Data mining is mining proceses or extracting information by getting pattern or specific rules from large amounts of data that can solve those condition. Taking advantage of student data and student graduation data, expected to yield information about the relationship with a graduation rate of students holding the data through data mining technique. The graduation rate is measured from the time of study and GPA. Data mining techniques use apriori algorithm, the information displayed in the form of support and confidence values from each graduation rate category.

Keywords: data mining, apriori algorithm, graduation rates, student master data.

KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul ”Aplikasi *Data Mining* Untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa (Studi Kasus di Fakultas MIPA Universitas Diponegoro)”.

Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana strata satu pada Program Studi Teknik Informatika Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro Semarang.

Dalam penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak, oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Ibu Dra. Rum Hastuti, M.Si selaku Dekan Fakultas MIPA Universitas Diponegoro.
2. Ibu Dr. Widowati, S.Si, M.Si selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Diponegoro.
3. Bapak Drs. Eko Adi Sarwoko, M.Kom selaku Ketua Program Studi Teknik Informatika Jurusan Matematika Fakultas MIPA Universitas Diponegoro.
4. Bapak Priyo Sidik Sasongko, S.Si, M.Kom selaku koordinator tugas akhir.
5. Bapak Aris Puji Widodo, S.Si, M.T. dan Bapak Helmie Arif Wibawa, S.Si, M.Cs, selaku dosen wali.
6. Ibu Beta Noranita, S.Si, M.Kom selaku dosen pembimbing I yang telah membantu dalam membimbing dan mengarahkan penulis.
7. Bapak Nurdin Bahtiar, S.Si, M.T, dan Ibu Awalina Kurniastuti, S.Si selaku dosen pembimbing II yang senantiasa meluangkan waktunya untuk memberikan bantuan, dukungan, arahan serta masukan bagi penulis dalam menyelesaikan tugas akhir ini.
8. Bapak Drs. Djalal Er Riyanto, M.IKomp, Bapak Drs. Kushartantya, M.IKomp, Bapak Drs. Suhartono, M.Kom, Bapak Edy S, S.T, dan seluruh dosen Program Studi Teknik Informatika Fakultas MIPA Universitas Diponegoro yang telah memberikan ilmu pengetahuan dan bimbingan kepada penulis.
9. Bapak Mauludi, Bapak Reynaldi, Bapak Warto, Ibu tammy, dan seluruh staf bagian akademik Fakultas MIPA Universitas Diponegoro atas segala bantuan dan kerjasamanya.

10. Ayah, Ibu, serta seluruh keluarga tercinta yang selalu memberikan doa kasih sayang, dukungan, semangat dan segalanya yang tiada henti-hentinya kepada penulis.
11. Muh. Abdurrahman, S. Kom, Fajar K, S.Kom, Yanuar, Endah Etrisari, S.Kom, Rr. Hernitasari, Azizah, S.Kom, Ruli Arista, S.Kom, Teguh, Nurrohman, Herny, Fera, Wahyu dan seluruh teman-teman Teknik Informatika yang telah banyak membantu segala hal dalam penyusunan laporan tugas akhir ini.
12. Xavi, Bowo, dan seluruh temen satu kos, yang banyak membantu.
13. Keluarga besar “*Alpharian Fotografer Club*” dan “Komunitas Fotografer Semarang” yang telah memberikan dorongan, dukungan, dan inspirasi.
14. Semua pihak yang telah membantu penulis yang tidak dapat disebutkan namanya satu persatu oleh penulis.

Penulis merasa masih banyak kekurangan dalam penyusunan laporan tugas akhir ini. Untuk itu, penulis mengharapkan saran dan kritik yang membangun dari pembaca. Semoga tugas akhir ini dapat bermanfaat, khususnya pada bidang Teknik Informatika.

Semarang, Agustus 2010

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
I. 1. Latar Belakang	1
I. 2. Rumusan Masalah	2
I. 3. Tujuan dan Manfaat	2
I. 4. Batasan Masalah.....	3
I. 5. Metode Pengambilan Data	4
I. 6. Sistematika Penulisan.....	4
BAB II DASAR TEORI.....	6
II. 1. <i>Data warehouse</i>	6
II. 2. Pengertian <i>Data mining</i>	10
II. 3. Pengenalan Pola, <i>Data mining</i> , dan <i>Machine Learning</i>	11
II. 4. Tahap-Tahap <i>Data mining</i>	12
II. 5. Metode <i>Data mining</i>	14
II. 5. 1. <i>Association rules</i>	14
II. 5. 2. <i>Decision Tree</i>	18
II. 5. 3. <i>Clustering</i>	19
II. 6. <i>Software Aplikasi</i>	20
II. 7. Basis Data dan Sistem Manajemen Basis Data (<i>Database and Database Management System</i>)	20
II. 8. Kamus Data (<i>Data Dictionary</i>)	22
II. 9. Desain Model Aplikasi.....	23
II. 10. Perancangan Perangkat Lunak	24
II. 11. Implementasi dan Pengujian Unit	25

BAB III ANALISIS DAN PERANCANGAN APLIKASI <i>DATA MINING</i>	26
III. 1. Analisis Data Mining	26
III. 1. 1. Sumber Data	27
III. 1. 2. Data Yang Digunakan.....	30
III. 1. 3. Integrasi Data.....	31
III. 1. 4. Transformasi Data.....	31
III. 1. 5. Penggunaan Algoritma <i>Apriori</i>	33
III. 1. 6. Report dan Penyajian Hasil Proses	39
III. 2. Analisis Lingkungan Sistem.....	39
III. 3. Analisis Perangkat Lunak.....	39
III. 3. 1. Deskripsi Umum Perangkat Lunak.....	39
III. 3. 2. Spesifikasi Kebutuhan Fungsional.....	40
III. 3. 3. Pemodelan Data	40
III. 3. 4. Pemodelan Fungsi.....	45
III. 4. Perancangan Perangkat Lunak	47
III. 4. 1. Perancangan Fungsi	47
III. 4. 2. Kebutuhan Antarmuka.....	50
III. 4. 3. Rancangan Tampilan	52
BAB IV IMPLEMENTASI PROGRAM DAN PENGUJIAN	54
IV. 1. Lingkungan Pembangunan	54
IV. 2. Implementasi Data.....	54
IV. 3. Implementasi Fungsi	56
IV. 4. Implementasi Rancangan Antarmuka	56
IV. 5. Pengujian Aplikasi <i>Data Mining</i>	58
IV. 5. 1. Lingkungan Pengujian	58
IV. 5. 2. Rancangan Pengujian.....	59
IV. 5. 3. Hasil Uji.....	59
IV. 5. 4. Analisis Hasil Uji.....	60
BAB V PENUTUP	61
V. 1. Kesimpulan.....	61
V. 2. Saran.....	61
DAFTAR PUSTAKA.....	62
LAMPIRAN 1	65

DAFTAR GAMBAR

Gambar 2.1 Kategori ETL Berdasarkan Siapa yang Menjalankan.	8
Gambar 2.2 Kategori ETL Berdasarkan Tempat Dijalankan	9
Gambar 2.3 Gambaran <i>Data Warehouse</i> Secara Sederhana	9
Gambar 2.4 <i>Data Mining</i> Merupakan Irisan Dari Berbagai Disiplin.	11
Gambar 2.5 Tahap-Tahap <i>Data Mining</i>	12
Gambar 2.6 <i>Decision Tree</i>	18
Gambar 2.7 <i>Clustering</i>	19
Gambar 3.1 Aliran Data Dalam Proses <i>Data Mining</i>	39
Gambar 3.2 Proses ETL	41
Gambar 3.3 Proses Pembangkitan ETL.....	41
Gambar 3.4 Proses <i>Load Data</i>	42
Gambar 3.5 DFD level-0	45
Gambar 3.6 DFD level-1	45
Gambar 3.7 Desain Tampilan Form Awal Aplikasi <i>Data Mining</i>	52
Gambar 3.8 Desain Tampilan Form Report <i>Data Mining</i>	53
Gambar 4.1 Tampilan Form Awal Aplikasi <i>Data Mining</i>	56
Gambar 4.2 Tampilan Form Report <i>Data Mining</i>	57

DAFTAR TABEL

Tabel 2.1 Simbol-Simbol <i>Data Dictionary</i>	22
Tabel 2.2 Simbol-Simbol <i>Context Diagram</i>	23
Tabel 2.3 Simbol-Simbol DFD.....	24
Tabel 3.1 Tabel Data Induk Mahasiswa	27
Tabel 3.2 Tabel Data Kelulusan	28
Tabel 3.3 Predikat Kelulusan.....	32
Tabel 3.4 Transformasi Data	32
Tabel 3.5 Data Awal	34
Tabel 3.6 Kandidat Pertama (C1)	34
Tabel 3.7 Hasil Setelah <i>Threshold</i> Ditetapkan (L1)	34
Tabel 3.8 Kandidat Kedua (C2).....	35
Tabel 3.9 Hasil Kedua (L2)	35
Tabel 3.10 Data Awal Contoh Kedua.....	36
Tabel 3.11 Kandidat Pertama (C1) Contoh Kedua.....	36
Tabel 3.12 Hasil Setelah <i>Threshold</i> Ditetapkan (L1) Contoh Kedua.....	37
Tabel 3.13 Kandidat Kedua (C2) Contoh Kedua	37
Tabel 3.14 Hasil Kedua (L2) Contoh Kedua.....	37
Tabel 3.15 Kandidat Ketiga (C3)	38
Tabel 3.16 Tabel Atribut Data Gabungan	43
Tabel 4.1 Struktur Tabel Data Gabungan.....	55
Tabel 4.2 Identifikasi dan Pelaksanaan Pengujian	59
Tabel 4.3 Hasil Uji Aplikasi <i>Data Mining</i>	60

BAB I

PENDAHULUAN

I. 1. Latar Belakang

Dengan kemajuan teknologi informasi dewasa ini, kebutuhan akan informasi yang akurat sangat dibutuhkan dalam kehidupan sehari-hari, sehingga informasi akan menjadi suatu elemen penting dalam perkembangan masyarakat saat ini dan waktu mendatang. Namun kebutuhan informasi yang tinggi kadang tidak diimbangi dengan penyajian informasi yang memadai, sering kali informasi tersebut masih harus di gali ulang dari data yang jumlahnya sangat besar. Kemampuan teknologi informasi untuk mengumpulkan dan menyimpan berbagai tipe data jauh meninggalkan kemampuan untuk menganalisis, meringkas dan mengekstrak pengetahuan dari data. Metode tradisional untuk menganalisis data yang ada, tidak dapat menangani data dalam jumlah besar.

Pemanfaatan data yang ada di dalam sistem informasi untuk menunjang kegiatan pengambilan keputusan, tidak cukup hanya mengandalkan data operasional saja, diperlukan suatu analisis data untuk menggali potensi-potensi informasi yang ada. Para pengambil keputusan berusaha untuk memanfaatkan gudang data yang sudah dimiliki untuk menggali informasi yang berguna membantu mengambil keputusan, hal ini mendorong munculnya cabang ilmu baru untuk mengatasi masalah penggalian informasi atau pola yang penting atau menarik dari data dalam jumlah besar, yang disebut dengan *data mining*. Penggunaan teknik *data mining* diharapkan dapat memberikan pengetahuan-pengetahuan yang sebelumnya tersembunyi di dalam gudang data sehingga menjadi informasi yang berharga.

Perguruan tinggi saat ini dituntut untuk memiliki keunggulan bersaing dengan memanfaatkan semua sumber daya yang dimiliki. Selain sumber daya sarana, prasarana, dan manusia, sistem informasi adalah salah satu sumber daya yang dapat digunakan untuk meningkatkan keunggulan bersaing. Sistem informasi dapat digunakan untuk mendapatkan, mengolah dan menyebarkan informasi untuk menunjang kegiatan operasional sehari-hari sekaligus menunjang kegiatan pengambilan keputusan strategis.

Di dalam peraturan akademik Universitas Diponegoro bidang pendidikan tahun 2009 pada BAB I pasal 1 ayat 2 disebutkan bahwa “Program Sarjana (S1) reguler adalah program pendidikan akademik setelah pendidikan menengah, yang memiliki beban studi sekurang-kurangnya 144 (seratus empat puluh empat) sks (satuan kredit semester) dan sebanyak-banyaknya 160 (seratus enam puluh) sks yang dijadwalkan untuk 8 (delapan) semester dan dapat ditempuh dalam waktu kurang dari 8 (delapan) semester dan paling lama 14 (empat belas) semester” (Peraturan Akademik, 2009). Berdasarkan buku wisuda angkatan ke-115, 48 dari 80 peserta wisuda Program Sarjana (S1) reguler di Fakultas Matematika dan Ilmu Pengetahuan Alam (MIPA) menempuh masa studi lebih dari 8 semester. Hal ini menunjukkan bahwa masih banyak mahasiswa Program Sarjana (S1) reguler di Fakultas MIPA yang menempuh lama studi lebih dari 8 semester dari yang dijadwalkan 8 semester. Oleh karena itu, dengan memanfaatkan data induk mahasiswa dan data kelulusan mahasiswa, dapat diketahui informasi tingkat kelulusan mahasiswa melalui teknik *data mining*.

I. 2. Rumusan Masalah

Tingkat kelulusan mahasiswa dapat dilihat dari lama studi dan IPK (Indeks Prestasi Kumulatif) yang terdapat pada data kelulusan mahasiswa. *Data mining* diharapkan dapat membantu menyajikan informasi tentang tingkat kelulusan mahasiswa dengan menggunakan data kelulusan mahasiswa dan data induk mahasiswa.

Permasalahan yang dibahas dalam tugas akhir ini adalah bagaimana membuat aplikasi untuk menghasilkan informasi yang berguna tentang hubungan tingkat kelulusan dengan data induk mahasiswa dengan teknik *data mining*. Informasi yang ditampilkan berupa nilai *support* dan *confidence* hubungan antara tingkat kelulusan dengan data induk mahasiswa

I. 3. Tujuan dan Manfaat

Tujuan yang ingin dicapai dari pelaksanaan dan penulisan tugas akhir ini adalah menghasilkan aplikasi untuk mendapatkan informasi yang berguna tentang tingkat kelulusan mahasiswa dengan teknik *data mining*.

Adapun beberapa manfaat yang diharapkan pada pembuatan tugas akhir ini adalah:

1. Bagi Penulis

Penulis dapat lebih mengetahui cara menerapkan ilmu-ilmu yang telah dipelajari selama ini dalam merancang dan membuat aplikasi sistem dengan teknik *data mining*, serta sebagai syarat dalam memperoleh gelar sarjana komputer.

2. Bagi Fakultas MIPA

Diharapkan dengan adanya aplikasi ini dapat membantu menyajikan informasi tentang hubungan tingkat kelulusan dengan data induk mahasiswa. Pihak fakultas dapat mengetahui tingkat kelulusan mahasiswanya dan mengetahui faktor yang mempengaruhi tingkat kelulusan.

I. 4. Batasan Masalah

Pada tugas akhir ini, pembahasan dibatasi pada menyajikan informasi tentang tingkat kelulusan mahasiswa dengan teknik *data mining*. Informasi yang ditampilkan berupa nilai *support* dan *confidence* hubungan antara tingkat kelulusan dengan data induk mahasiswa. Dalam penulisan tugas akhir ini tidak membahas pada sistem pendukung keputusan maupun sistem informasi akademik. Dalam membangun *data mining* membutuhkan suatu *data warehouse*, oleh karena itu dalam pembahasan tugas akhir ini dibahas mengenai pembangunan *data warehouse* sederhana yang dibangun untuk memenuhi kebutuhan dari proses *data mining*. *Data warehouse* yang dibangun bukan merupakan *data warehouse* yang menyimpan seluruh data transaksional, hanya merupakan *data warehouse* yang menunjang pembangunan *data mining*, sehingga data dan formatnya pun disesuaikan dengan kebutuhan *data mining*.

Pembahasan juga dibatasi pada bagaimana menghasilkan aplikasi yang menerapkan teknik *data mining* guna menghasilkan informasi hubungan tingkat kelulusan dengan data induk mahasiswa. Dalam tugas akhir ini tidak membahas pada hasil proses *data mining* dan analisis hasil yang keluar. Pembahasan juga hanya pada Program Sarjana (S1) reguler di Fakultas MIPA Universitas Diponegoro secara umum. Data yang diambil adalah data mahasiswa untuk

Program Sarjana (S1) reguler di Fakultas MIPA Universitas Diponegoro. Data induk mahasiswa adalah atribut yang melekat pada mahasiswa seperti nama, NIM (Nomor Induk Mahasiswa), alamat, asal sekolah, dan lain-lain. Tingkat kelulusan diukur dari lama studi dan IPK. Dalam Penulisan tugas akhir ini, lama studi dan IPK mengacu pada peraturan akademik tahun 2009 nomor : 364/PER/H7/2009 tanggal 24 Juli 2009. Lama studi dikategorikan berdasarkan peraturan akademik BAB I pasal 1 ayat 2, sedangkan IPK dikategorikan berdasarkan predikat kelulusan yang diatur dalam peraturan akademik BAB IV pasal 19 ayat 1. Implementasi program menggunakan *SQL Server* 2005 dan *Visual Studio* 2010 dengan bahasa pemrograman *Visual Basic. Net*.

I. 5. Metode Pengambilan Data

Metode yang digunakan dalam pengumpulan data adalah sebagai berikut :

1. Metode Pengamatan Langsung

Melakukan pengamatan langsung ke bagian Akademik Fakultas MIPA untuk mendapatkan data yang dibutuhkan.

2. Metode Wawancara

Mengadakan wawancara dengan pihak-pihak yang berkaitan langsung dengan permasalahan yang sedang dibahas pada tugas akhir ini untuk memperoleh gambaran dan penjelasan secara mendasar.

3. Metode Studi Pustaka

Merupakan sumber yang dapat dijadikan rujukan dari sumber data atau literatur-literatur.

4. Metode Browsing

Melakukan pengumpulan rujukan yang bersumber dari internet.

I. 6. Sistematika Penulisan

Sistematika dari penulisan tugas sarjana ini adalah sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang permasalahan, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II DASAR TEORI

Pada bab ini dijelaskan mengenai dasar-dasar teori, rujukan dan metode yang digunakan sebagai dasar dan alat untuk menyelesaikan permasalahan.

BAB III ANALISIS DAN PERANCANGAN APLIKASI *DATA MINING*

Pada bab ini dijelaskan tentang analisis serta perancangan Aplikasi *Data Mining*.

BAB IV IMPLEMENTASI PROGRAM DAN PENGUJIAN

Pada bab ini berisi penerapan teknik *data mining* dalam aplikasi, pembuatan *prototype* Aplikasi *Data Mining* dan pengujian.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari hasil pembuatan Aplikasi *Data Mining* dan saran-saran yang ditujukan kepada semua pihak yang bersangkutan.

BAB II

DASAR TEORI

Bab ini menjelaskan tentang dasar teori yang digunakan dalam penyusunan tugas akhir ini. Dijelaskan pengertian tentang *data mining* beserta macam-macamnya, selain itu juga dijelaskan tentang *data warehouse*, *database* serta analisis perancangan perangkat lunak.

II. 1. *Data warehouse*

Data warehouse adalah sebuah sistem yang mengambil dan menggabungkan data secara periodik dari sistem sumber data ke penyimpanan data bentuk dimensional atau normal (Rainardi, 2008). *Data warehouse* merupakan penyimpanan data yang berorientasi objek, terintegrasi, mempunyai *variant* waktu, dan menyimpan data dalam bentuk *nonvolatile* sebagai pendukung manajemen dalam proses pengambilan keputusan (Han, 2006).

Data warehouse menyatukan dan menggabungkan data dalam bentuk multidimensi. Pembangunan *data warehouse* meliputi pembersihan data, penyatuan data dan transformasi data dan dapat dilihat sebagai praproses yang penting untuk digunakan dalam data mining. Selain itu *data warehouse* mendukung *On-line Analytical Processing* (OLAP), sebuah kakas yang digunakan untuk menganalisis secara interaktif dari bentuk multidimensi yang mempunyai data yang rinci. Sehingga dapat memfasilitasi secara efektif data *generalization* dan *data mining*.

Banyak metode-metode *data mining* yang lain seperti asosiasi, klasifikasi, prediksi, dan *clustering*, dapat diintegrasikan dengan operasi OLAP untuk meningkatkan proses *mining* yang interaktif dari beberapa level dari abstraksi. Oleh karena itu *data warehouse* menjadi platform yang penting untuk data analisis dan OLAP untuk dapat menyediakan platform yang efektif untuk proses *data mining*.

Empat karakteristik dari *data warehouse* meliputi :

1. *Subject oriented* : sebuah *data warehouse* disusun dalam subjek utama, seperti pelanggan, suplier, produk, dan sales. Meskipun *data warehouse* terkonsentrasi pada operasi harian dan proses transaksi dalam perusahaan, *data warehouse*

fokus pada pemodelan dan analisis data untuk pembuat keputusan. Oleh karena itu *data warehouse* mempunyai karakter menyediakan secara singkat dan sederhana gambaran seputar subjek lebih detail yang dibuat dari data luar yang tidak berguna dalam proses pendukung keputusan.

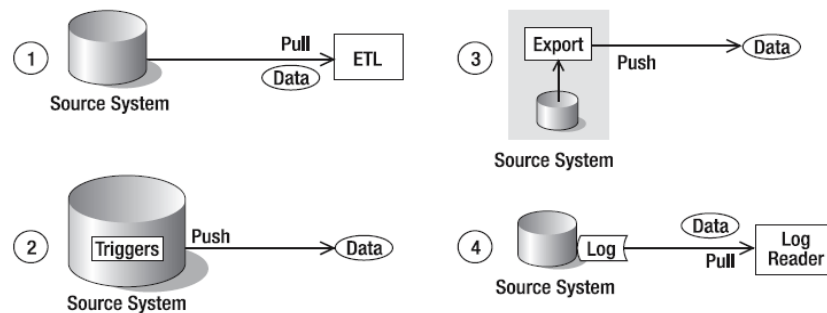
2. *Integrated* : *Data warehouse* biasanya dibangun dari bermacam-macam sumber yang berbeda, seperti *database* relasional, *flat files*, dan *on-line transaction records*. Pembersihan dan penyatuan data diterapkan untuk menjamin konsistensi dalam penamaan, struktur kode, ukuran atribut, dan yang lainnya.
3. *Time Variant* : data disimpan untuk menyajikan informasi dari sudut pandang masa lampau (misal 5 – 10 tahun yang lalu). Setiap struktur kunci dalam *data warehouse* mempunyai elemen waktu baik secara implisit maupun eksplisit
4. *Nonvolatile* : sebuah *data warehouse* secara fisik selalu disimpan terpisah dari data aplikasi operasional. Penyimpanan yang terpisah ini, *data warehouse* tidak memerlukan proses transaksi, *recovery* dan mekanisme pengendalian konkurensi. Biasanya hanya membutuhkan dua operasi dalam akses data yaitu *initial load of data* dan *access of data*

Dari pengertian tersebut, sebuah *data warehouse* merupakan penyimpanan data tetap sebagai implementasi fisik dari pendukung keputusan model data. *Data warehouse* juga biasanya dilihat sebagai arsitektur, pembangunan dan penyatuan data dari bermacam macam sumber data yang berbeda untuk mendukung struktur dan atau *query* tertentu, laporan analisis, dan pembuatan keputusan (Han, 2006).

Extract, transform, dan load (ETL) merupakan sebuah sistem yang dapat membaca data dari suatu *data store*, merubah bentuk data, dan menyimpan ke *data store* yang lain. *Data store* yang dibaca ETL disebut *data source*, sedangkan *data store* yang disimpan ETL disebut target. Proses pengubahan data digunakan agar data sesuai dengan format dan kriteria, atau sebagai validasi data dari *source system*. Proses ETL tidak hanya menyimpan data ke *data warehouse*, tetapi juga digunakan untuk berbagai proses pemindahan data. Kebanyakan ETL mempunyai mekanisme untuk membersihkan data dari *source system* sebelum disimpan ke *warehouse*. Pembersihan data merupakan proses identifikasi dan koreksi data yang kotor. Proses pembersihan ini menerapkan aturan-aturan tertentu yang mendefinisikan data bersih.

Berdasarkan siapa yang memindahkan data, ETL dapat dibedakan menjadi empat seperti yang dapat dilihat pada gambar 2.1, yaitu :

1. Proses ETL menarik data keluar dengan *query* tertentu di *source system database* secara periodik.
2. *Triggers* pada *source system* mendorong data keluar. *Triggers* adalah Suatu SQL statement yang dijalankan setiap ada perintah insert, update, atau delete dalam tabel.
3. Penjadwalan proses dalam *source system* untuk mengekspor data secara periodik. Hal ini mirip dengan proses yang pertama namun *query* disimpan dalam *data source*.
4. Sebuah log reader yang bertugas membaca log dalam *source system* untuk mengidentifikasi perubahan data. Log reader merupakan program yang membaca log file. Setelah dibaca, kemudian data dipindahkan keluar ke tempat penyimpanan yang lain.



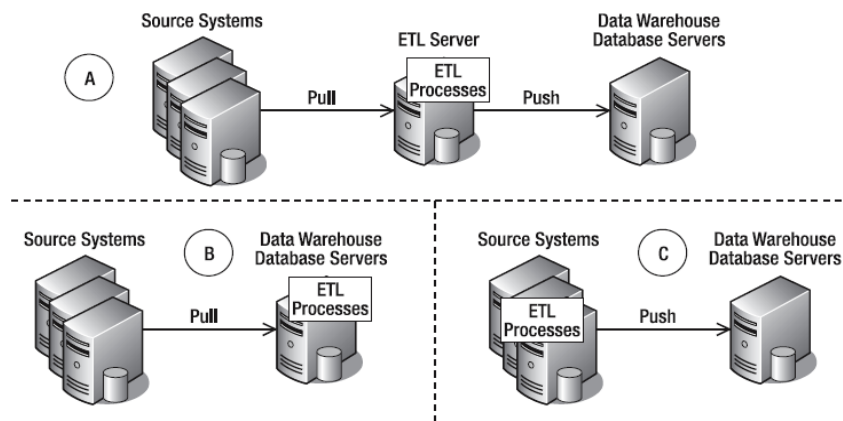
Gambar 2.1 Kategori ETL berdasarkan siapa yang menjalankan.

Berdasarkan dimana proses pembangkitan ETL, ETL dibedakan menjadi tiga macam seperti yang digambarkan pada gambar 2.2, yaitu :

- A. ETL dijalankan dalam *server* terpisah diantara *source system* dan *data warehouse* sistem. Pendekatan ini menghasilkan kinerja tinggi, ETL berjalan di *server* sendiri, sehingga tidak menggunakan sumber daya dari *data warehouse server* atau *data source server*. Namun hal ini lebih mahal karena harus menambah *server* lagi.
- B. ETL dijalankan dalam *data warehouse server*. Pendekatan ini dapat digunakan jika mempunyai kapasitas lebih dalam *data warehouse server* atau jika mempunyai *idle time* ketika *data warehouse* tidak digunakan (misal pada

waktu malam). Pendekatan ini lebih murah dibandingkan pendekatan pertama karena tidak membutuhkan tambahan *server*.

- C. ETL dijalankan pada *server data source*. Pendekatan ini diimplementasikan ketika membutuhkan *real time data warehousing*. Dengan kata lain, jika data dalam *source system* berubah, perubahan ini dilakukan juga ke dalam *data warehouse*. Hal ini dapat dilakukan dengan penggunaan trigger dalam *source system*.



Gambar 2.2 Kategori ETL berdasarkan tempat dijalankan

Tidak semua *data warehouse* mempunyai komponen lengkap seperti mekanisme kualitas data, *database* multidimensi, aplikasi analisis, aplikasi pengguna, control sistem, audit sistem, metadata. Secara sederhana *data warehouse* dapat digambarkan seperti gambar 2.3



Gambar 2.3 Gambaran *data warehouse* secara sederhana

Dalam hal ini, *data warehouse* hanya mempunyai sebuah ETL dan sebuah *data store*. *Source system* bukan merupakan bagian dari *data warehouse* sistem. Hal ini merupakan minimum dari sebuah *data warehouse*. Jika satu komponen diambil sudah bukan merupakan *data warehouse* lagi (Rainardi, 2008).

II. 2. Pengertian *Data Mining*

Secara sederhana *data mining* adalah penambangan atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar (Davies, 2004). *Data mining* juga disebut sebagai serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data (Pramudiono, 2007). *Data mining*, sering juga disebut sebagai *knowledge discovery in database* (KDD). KDD adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar (Santoso, 2007).

Data mining adalah kegiatan menemukan pola yang menarik dari data dalam jumlah besar, data dapat disimpan dalam *database*, *data warehouse*, atau penyimpanan informasi lainnya. *Data mining* berkaitan dengan bidang ilmu – ilmu lain, seperti *database system*, *data warehousing*, statistik, *machine learning*, *information retrieval*, dan komputasi tingkat tinggi. Selain itu, *data mining* didukung oleh ilmu lain seperti *neural network*, pengenalan pola, *spatial data analysis*, *image database*, *signal processing* (Han, 2006). *Data mining* didefinisikan sebagai proses menemukan pola-pola dalam data. Proses ini otomatis atau seringnya semiotomatis. Pola yang ditemukan harus penuh arti dan pola tersebut memberikan keuntungan, biasanya keuntungan secara ekonomi. Data yang dibutuhkan dalam jumlah besar (Witten, 2005).

Karakteristik *data mining* sebagai berikut

- *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
- *Data mining* biasa menggunakan data yang sangat besar. Biasanya data yang besar digunakan untuk membuat hasil lebih dipercaya.
- *Data mining* berguna untuk membuat keputusan yang kritis, terutama dalam strategi (Davies, 2004).

Berdasarkan beberapa pengertian tersebut dapat ditarik kesimpulan bahwa *data mining* adalah suatu teknik menggali informasi berharga yang terpendam atau tersembunyi pada suatu koleksi data (*database*) yang sangat besar sehingga ditemukan suatu pola yang menarik yang sebelumnya tidak diketahui. Kata

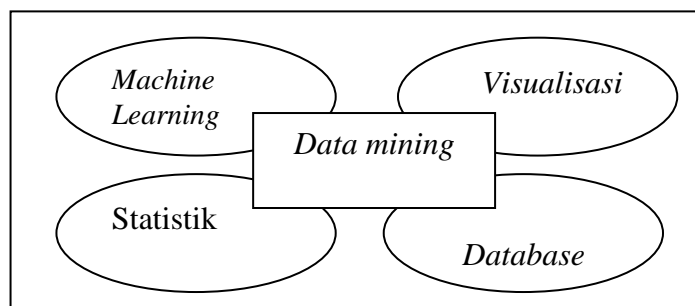
mining sendiri berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar. Karena itu *data mining* sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistik dan *database*. Beberapa metode yang sering disebut-sebut dalam literatur *data mining* antara lain *clustering*, *classification*, *association rules mining*, *neural network*, *genetic algorithm* dan lain-lain (Pramudiono, 2007).

II. 3. Pengenalan Pola, *Data Mining*, dan *Machine Learning*

Pengenalan pola adalah suatu disiplin ilmu yang mempelajari cara-cara mengklasifikasikan obyek ke beberapa kelas atau kategori dan mengenali kecenderungan data. Tergantung pada aplikasinya, obyek-obyek ini bisa berupa pasien, mahasiswa, pemohon kredit, *image* atau *signal* atau pengukuran lain yang perlu diklasifikasikan atau dicari fungsi regresinya (Santoso, 2007).

Data mining, sering juga disebut *knowledge discovery in database* (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan. Sehingga istilah *pattern recognition* jarang digunakan karena termasuk bagian dari *data mining* (Santoso, 2007).

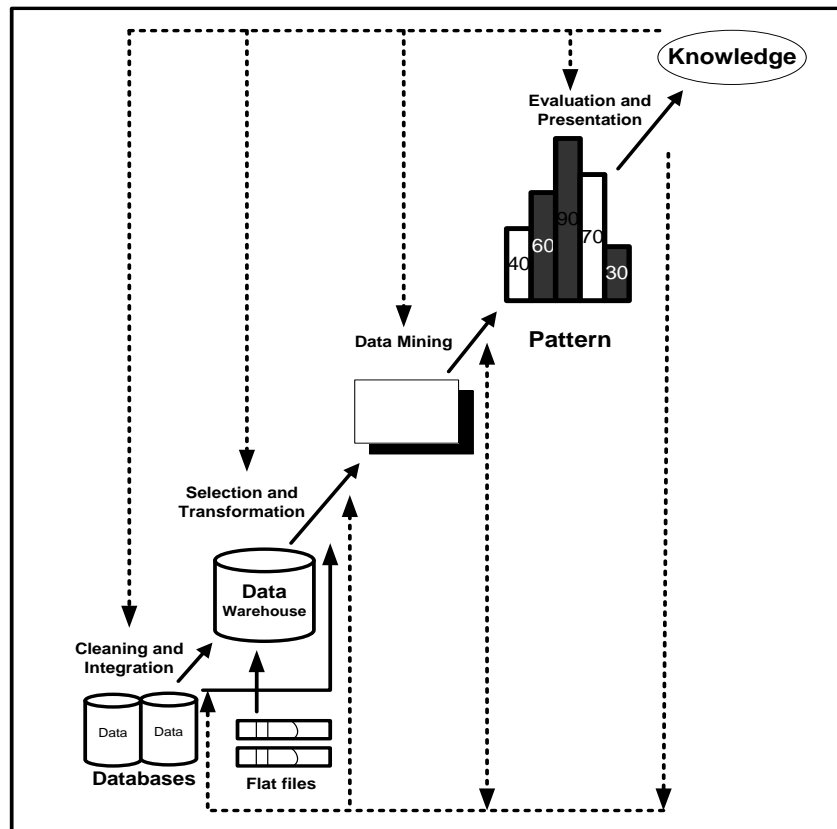
Machine Learning adalah suatu area dalam *artificial intelligence* atau kecerdasan buatan yang berhubungan dengan pengembangan teknik-teknik yang bisa diprogramkan dan belajar dari data masa lalu. Pengenalan pola, *data mining* dan *machine learning* sering dipakai untuk menyebut sesuatu yang sama. Bidang ini bersinggungan dengan ilmu probabilitas dan statistik kadang juga optimasi. *Machine learning* menjadi alat analisis dalam *data mining*. Bagaimana bidang-bidang ini berhubungan bisa dilihat dalam gambar 2.4 (Santoso, 2007).



Gambar 2.4 *Data mining* merupakan irisan dari berbagai disiplin.

II. 4. Tahap-Tahap *Data mining*

Sebagai suatu rangkaian proses, *data mining* dapat dibagi menjadi beberapa tahap yang diilustrasikan di Gambar 2.5. Tahap-tahap tersebut bersifat interaktif, pemakai terlibat langsung atau dengan perantaraan *knowledge base*.



Gambar 2.5 Tahap-Tahap *Data Mining* (Han, 2006).

Tahap-tahap *data mining* ada 6 yaitu :

1. Pembersihan data (*data cleaning*)

Pembersihan data merupakan proses menghilangkan noise dan data yang tidak konsisten atau data tidak relevan. Pada umumnya data yang diperoleh, baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa *data mining* yang dimiliki. Data-data yang tidak relevan itu juga lebih baik dibuang. Pembersihan data juga akan mempengaruhi performansi dari teknik *data mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Integrasi data (*data integration*)

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Tidak jarang data yang diperlukan untuk *data mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau file teks. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasi entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

3. Seleksi Data (*Data Selection*)

Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*. Sebagai contoh, sebuah kasus yang meneliti faktor kecenderungan orang membeli dalam kasus *market basket analysis*, tidak perlu mengambil nama pelanggan, cukup dengan id pelanggan saja.

4. Transformasi data (*Data Transformation*)

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*. Beberapa metode *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa metode standar seperti analisis asosiasi dan *clustering* hanya bisa menerima input data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagi-bagi menjadi beberapa interval. Proses ini sering disebut transformasi data.

5. Proses *mining*,

Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

6. Evaluasi pola (*pattern evaluation*),

Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan. Dalam tahap ini hasil dari teknik *data mining* berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada

memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai hipotesa ada beberapa alternatif yang dapat diambil seperti menjadikannya umpan balik untuk memperbaiki proses *data mining*, mencoba metode *data mining* lain yang lebih sesuai, atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

7. Presentasi pengetahuan (*knowledge presentation*),

Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Tahap terakhir dari proses *data mining* adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisis yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak memahami *data mining*. Karenanya presentasi hasil *data mining* dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses *data mining*. Dalam presentasi ini, visualisasi juga bisa membantu mengkomunikasikan hasil *data mining* (Han, 2006).

II. 5. Metode *Data mining*

Dengan definisi *data mining* yang luas, ada banyak jenis metode analisis yang dapat digolongkan dalam *data mining*.

II. 5. 1. *Association rules*

Association rules (aturan asosiasi) atau *affinity analysis* (analisis afinitas) berkenaan dengan studi tentang “apa bersama apa”. Sebagai contoh dapat berupa studi transaksi di supermarket, misalnya seseorang yang membeli susu bayi juga membeli sabun mandi. Pada kasus ini berarti susu bayi bersama dengan sabun mandi. Karena awalnya berasal dari studi tentang *database* transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama produk apa, maka aturan asosiasi juga sering dinamakan *market basket analysis*.

Aturan asosiasi ingin memberikan informasi tersebut dalam bentuk hubungan “if-then” atau “jika-maka”. Aturan ini dihitung dari data yang sifatnya probabilistik (Santoso, 2007).

Analisis asosiasi dikenal juga sebagai salah satu metode *data mining* yang menjadi dasar dari berbagai metode *data mining* lainnya. Khususnya salah satu

tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi (*frequent pattern mining*) menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien. Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua parameter, *support* (nilai penunjang) yaitu prosentase kombinasi item tersebut dalam *database* dan *confidence* (nilai kepastian) yaitu kuatnya hubungan antar item dalam aturan assosiatif. Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan assosiatif yang memenuhi syarat minimum untuk *support* (*minimum support*) dan syarat minimum untuk *confidence* (*minimum confidence*) (Pramudiono, 2007).

Ada beberapa algoritma yang sudah dikembangkan mengenai aturan asosiasi, namun ada satu algoritma klasik yang sering dipakai yaitu algoritma *apriori*. Ide dasar dari algoritma ini adalah dengan mengembangkan *frequent itemset*. Dengan menggunakan satu item dan secara rekursif mengembangkan *frequent itemset* dengan dua item, tiga item dan seterusnya hingga *frequent itemset* dengan semua ukuran.

Untuk mengembangkan *frequent set* dengan dua item, dapat menggunakan *frequent set item*. Alasannya adalah bila set satu item tidak melebihi *support minimum*, maka sembarang ukuran itemset yang lebih besar tidak akan melebihi *support minimum* tersebut. Secara umum, mengembangkan set dengan k -item menggunakan frequent set dengan $k - 1$ item yang dikembangkan dalam langkah sebelumnya. Setiap langkah memerlukan sekali pemeriksaan ke seluruh isi *database*.

Dalam asosiasi terdapat istilah *antecedent* dan *consequent*, *antecedent* untuk mewakili bagian “jika” dan *consequent* untuk mewakili bagian “maka”. Dalam analisis ini, *antecedent* dan *consequent* adalah sekelompok item yang tidak punya hubungan secara bersama (Santoso, 2007).

Dari jumlah besar aturan yang mungkin dikembangkan, perlu memiliki aturan-aturan yang cukup kuat tingkat ketergantungan antar item dalam *antecedent* dan *consequent*. Untuk mengukur kekuatan aturan asosiasi ini, digunakan ukuran *support* dan *confidence*. *Support* adalah rasio antara jumlah transaksi yang memuat *antecedent* dan *consequent* dengan jumlah transaksi. *Confidence* adalah rasio antara

jumlah transaksi yang meliputi semua item dalam *antecedent* dan *consequent* dengan jumlah transaksi yang meliputi semua item dalam *antecedent*.

$$S = \frac{\Sigma (Ta+Tc)}{\Sigma(T)} \dots\dots\dots(2.1)$$

Keterangan :

S = *Support*

$\Sigma (Ta + Tc)$ = Jumlah transaksi yang mengandung *antecedent* dan *consequencent*

$\Sigma(T)$ = Jumlah transaksi

$$C = \frac{\Sigma (Ta+Tc)}{\Sigma(Ta)} \dots\dots\dots(2.2)$$

Keterangan :

C = *Confidence*

$\Sigma (Ta + Tc)$ = Jumlah transaksi yang mengandung *antecedent* dan *consequencent*

$\Sigma(Ta)$ = Jumlah transaksi yang mengandung *antecedent*

Langkah pertama algoritma *apriori* adalah, *support* dari setiap item dihitung dengan men-scan *database*. Setelah *support* dari setiap item didapat, item yang memiliki *support* lebih besar dari *minimum support* dipilih sebagai pola frekuensi tinggi dengan panjang 1 atau sering disingkat 1-itemset. Singkatan k-itemset berarti satu set yang terdiri dari k item.

Iterasi kedua menghasilkan 2-itemset yang tiap set-nya memiliki dua item. Pertama dibuat kandidat 2-itemset dari kombinasi semua 1-itemset. Lalu untuk tiap kandidat 2-itemset ini dihitung *support*-nya dengan men-scan *database*. *Support* artinya jumlah transaksi dalam *database* yang mengandung kedua item dalam kandidat 2-itemset. Setelah *support* dari semua kandidat 2-itemset didapatkan, kandidat 2-itemset yang memenuhi syarat *minimum support* dapat ditetapkan sebagai 2-itemset yang juga merupakan pola frekuensi tinggi dengan panjang 2. (Pramudiono, 2007)

Untuk selanjutnya iterasi iterasi ke-k dapat dibagi lagi menjadi beberapa bagian :

1. Pembentukan kandidat itemset

Kandidat k-itemset dibentuk dari kombinasi (k-1)-itemset yang didapat dari iterasi sebelumnya. Satu ciri dari algoritma *apriori* adalah adanya pemangkasan kandidat k-itemset yang subset-nya yang berisi k-1 item tidak termasuk dalam pola frekuensi tinggi dengan panjang k-1.

2. Penghitungan support dari tiap kandidat k-itemset

Support dari tiap kandidat k-itemset didapat dengan men-scan *database* untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat k-itemset tersebut. Ini adalah juga ciri dari algoritma *apriori* yaitu diperlukan penghitungan dengan scan seluruh *database* sebanyak k-itemset terpanjang.

3. Tetapkan pola frekuensi tinggi

Pola frekuensi tinggi yang memuat k item atau k-itemset ditetapkan dari kandidat k-itemset yang support-nya lebih besar dari *minimum support*. Kemudian dihitung *confidence* masing-masing kombinasi item.

Iterasi berhenti ketika semua item telah dihitung sampai tidak ada kombinasi item lagi. (Pramudiono, 2007)

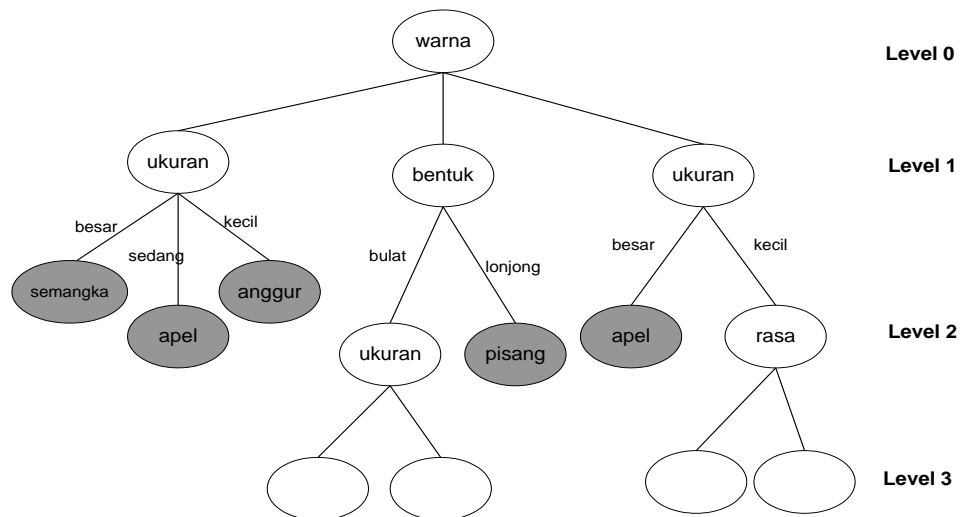
Secara ringkas algoritma *apriori* sebagai berikut :

```
Create L1 = set of supported itemsets of cardinality one
Set k to 2
while (Lk-1 ≠ ∅) {
    Create Ck from Lk-1
    Prune all the itemsets in Ck that are not
    supported, to create Lk
    Increase k by 1
}
The set of all supported itemsets is L1 ∪ L2 ∪ ... ∪ Lk
```

Selain algoritma *apriori*, terdapat juga algoritma lain seperti *FP-Grwoth*. Perbedaan algoritma *apriori* dengan *FP-Growth* pada banyaknya *scan database*. Algoritma *apriori* melakukan *scan database* setiap kali iterasi sedangkan algoritma *FP-Growth* hanya melakukan sekali di awal (Bramer, 2007).

II. 5. 2. *Decision Tree*

Dalam *decision tree* tidak menggunakan *vector* jarak untuk mengklasifikasikan obyek. Seringkali data observasi mempunyai atribut-atribut yang bernilai nominal. Seperti yang diilustrasikan pada gambar 2.6, misalkan obyeknya adalah sekumpulan buah-buahan yang bisa dibedakan berdasarkan atribut bentuk, warna, ukuran dan rasa. Bentuk, warna, ukuran dan rasa adalah besaran nominal, yaitu bersifat kategoris dan tiap nilai tidak bisa dijumlahkan atau dikurangkan. Dalam atribut warna ada beberapa nilai yang mungkin yaitu hijau, kuning, merah. Dalam atribut ukuran ada nilai besar, sedang dan kecil. Dengan nilai-nilai atribut ini, kemudian dibuat *decision tree* untuk menentukan suatu obyek termasuk jenis buah apa jika nilai tiap-tiap atribut diberikan (Santoso, 2007).



Gambar 2. 6 *Decision Tree*

Ada beberapa macam algoritma *decision tree* diantaranya CART dan C4.5. Beberapa isu utama dalam *decision tree* yang menjadi perhatian yaitu seberapa detail dalam mengembangkan *decision tree*, bagaimana mengatasi atribut yang bernilai *continues*, memilih ukuran yang cocok untuk penentuan atribut, menangani data training yang mempunyai data yang atributnya tidak mempunyai nilai, memperbaiki efisiensi perhitungan (Santoso, 2007).

Decision tree sesuai digunakan untuk kasus-kasus yang keluarannya bernilai diskrit. Walaupun banyak variasi model *decision tree* dengan tingkat kemampuan dan syarat yang berbeda, pada umumnya beberapa ciri yang cocok untuk diterapkannya *decision tree* adalah sebagai berikut :

1. Data dinyatakan dengan pasangan atribut dan nilainya
2. Label/keluaran data biasanya bernilai diskrit
3. Data mempunyai *missing value* (nilai dari suatu atribut tidak diketahui)

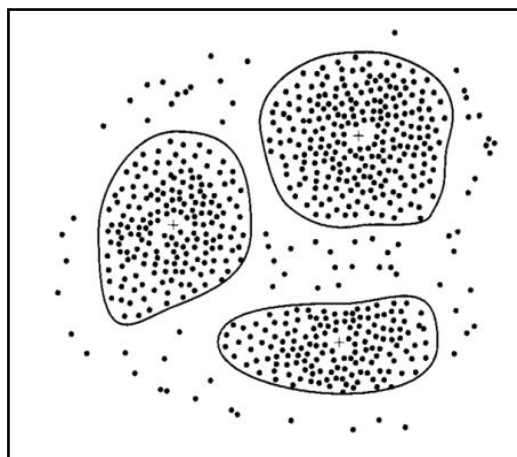
Dengan cara ini akan mudah mengelompokkan obyek ke dalam beberapa kelompok. Untuk membuat *decision tree* perlu memperhatikan hal-hal berikut ini :

1. Atribut mana yang akan dipilih untuk pemisahan obyek
2. Urutan atribut mana yang akan dipilih terlebih dahulu
3. Struktur *tree*
4. Kriteria pemberhentian
5. Pruning

(Santoso, 2007)

II. 5. 3. *Clustering*

Clustering termasuk metode yang sudah cukup dikenal dan banyak dipakai dalam *data mining*. Sampai sekarang para ilmuwan dalam bidang *data mining* masih melakukan berbagai usaha untuk melakukan perbaikan model *clustering* karena metode yang dikembangkan sekarang masih bersifat *heuristic*. Usaha-usaha untuk menghitung jumlah *cluster* yang optimal dan pengklasteran yang paling baik masih terus dilakukan. Dengan demikian menggunakan metode yang sekarang, tidak bisa menjamin hasil pengklasteran sudah merupakan hasil yang optimal. Namun, hasil yang dicapai biasanya sudah cukup bagus dari segi praktis.



Gambar 2.7 *Clustering*

Tujuan utama dari metode *clustering* adalah pengelompokan sejumlah data/obyek ke dalam *cluster (group)* sehingga dalam setiap *cluster* akan berisi data yang semirip mungkin seperti diilustrasikan pada gambar 2.7. Dalam *clustering*

metode ini berusaha untuk menempatkan obyek yang mirip (jaraknya dekat) dalam satu klaster dan membuat jarak antar klaster sejauh mungkin. Ini berarti obyek dalam satu *cluster* sangat mirip satu sama lain dan berbeda dengan obyek dalam *cluster-cluster* yang lain. Dalam metode ini tidak diketahui sebelumnya berapa jumlah *cluster* dan bagaimana pengelompokannya (Santoso, 2007).

II. 6. *Software Aplikasi*

Software aplikasi terdiri atas program yang berdiri sendiri yang mampu mengatasi kebutuhan bisnis tertentu. Aplikasi memfasilitasi operasi bisnis atau pengambilan keputusan manajemen maupun teknik sebagai tambahan dalam aplikasi pemrosesan data konvensional. *Software* aplikasi digunakan untuk mengatur fungsi bisnis secara *real time* (Pressman, 2005).

II. 7. *Basis Data dan Sistem Manajemen Basis Data (Database and Database Management System)*

Database adalah sekumpulan data yang saling berelasi (Elmasri, 2000). *Database* didesain, dibuat, dan diisi dengan data untuk tujuan mendapatkan informasi tertentu. Pendekatan *database* memiliki beberapa keuntungan seperti keberadaan katalog, independensi program-data, mendukung *view* (tampilan) untuk banyak pengguna, dan *sharing* data pada sejumlah transaksi. Selain itu masih ada fleksibilitas, ketersediaan *up-to-date* informasi untuk semua pengguna, skala ekonomis.

Kategori utama pengguna *database* terbagi menjadi empat kategori, yakni *Administrator*, *Designer*, *End user*, *System Analyst* dan *Application Programmers*. *Administrator* atau *Data Base Administrator* (DBA) bertanggung jawab pada otoritas akses *database*, koordinasi dan monitoring penggunaan, dan pemilihan perangkat keras dan lunak yang dibutuhkan. *Designer* bertanggung jawab pada identifikasi data yang disimpan dalam *database* dan memilih struktur yang tepat untuk menggambarkan dan menyimpan data. *End User* adalah orang yang kegiatannya membutuhkan akses ke *database* untuk melakukan *query*, *update*, dan membuat laporan. *System Analysts* menentukan kebutuhan *End User*. *Application Programmers* mengimplementasikan program sesuai spesifikasi.

Sistem Manajemen Basis Data (SMBD) adalah program yang digunakan pengguna untuk membuat dan memelihara *database*. SMBD memfasilitasi untuk mendefinisikan, mengkonstruksi, dan memanipulasi *database* untuk berbagai aplikasi. Pendefinisian *database* meliputi spesifikasi tipe data, struktur, dan constraint untuk data yang disimpan dalam *database*. Pengkonstruksian *database* adalah proses penyimpanan data itu sendiri pada media penyimpanan. Pemanipulasian *database* meliputi fungsi memanggil *query database* untuk mendapatkan data yang spesifik, *update database*, dan meng-*generate* laporan dari data tersebut (Elmasri, 2000).

Keuntungan yang diperoleh menggunakan SMBD meliputi mengontrol redundansi, membatasi akses yang tidak berwenang, menyediakan penyimpanan yang persisten, menghasilkan *interface* (antar muka) banyak pengguna, menjaga *integritas constraint*, menyediakan *backup* dan *recovery*.

Dalam SMBD menyediakan perintah yang digunakan untuk mengelola dan mengorganisasikan data, yakni *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). *Data Definition Language* adalah bahasa untuk mendefinisikan skema atau *database* fisik ke SMBD. (DDL). *Data Manipulation Language* adalah bahasa untuk memanipulasi data yaitu pengambilan informasi yang telah disimpan, penyisipan informasi baru, penghapusan informasi, modifikasi informasi yang disimpan dalam *database*. Selanjutnya, *query* adalah statemen yang ditulis untuk mengambil informasi. Bagian dari DML yang menangani pengambilan informasi ini disebut bahasa *query*.

SQL (dibaca "ess-que-el") singkatan dari *Structured Query Language*. SQL adalah bahasa yang digunakan untuk berkomunikasi dengan *database*. Menurut ANSI (*American National Standards Institute*), bahasa ini merupakan standard untuk *relational database management systems* (RDBMS).

Secara prinsip, perintah-perintah SQL (biasa disebut dengan pernyataan) dapat dibagi dalam tiga kelompok, yaitu :

- DDL (*Data Definition Language*) atau bahasa penerjemah data
Adalah perintah-perintah yang berkaitan dengan penciptaan atau penghapusan objek seperti tabel dan indek dalam *database*. Versi ANSI mencakup *CREATE TABLE*, *CREATE INDEX*, *ALTER TABLE*, *DROP TABLE*, *DROP VIEW*, dan

DROP INDEX. Beberapa sistem *database* menambahkan pernyataan DDL seperti *CREATE DATABASE* dan *CREATE SCHEMA*.

- **DML (*Data Manipulation Language*)** atau bahasa manipulasi data
Mencakup perintah-perintah yang digunakan untuk memanipulasi data. Misalnya untuk menambahkan data (*INSERT*), memperoleh data (*SELECT*), mengubah data (*UPDATE*), dan menghapus data (*DELETE*).
- **DCL (*Data Control Language*)** atau bahasa pengendali data
Merupakan kelompok perintah yang dipakai untuk melakukan otorisasi terhadap akses data dan pengalokasian ruang. Misalnya, suatu data bisa diakses si A, tetapi tidak bisa diakses oleh si B. Termasuk dalam kategori DCL yaitu pernyataan-pernyataan *GRANT*, *REVOKE*, *COMMIT*, dan *ROLLBACK* (Kadir, 1999)

II. 8. Kamus Data (*Data Dictionary*)

Kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau *file* dalam sistem.

Simbol-simbol yang digunakan dalam kamus data diterangkan dalam tabel 2.1.

Tabel 2.1 Simbol-Simbol *Data Dictionary*

Notasi	Arti
=	Terdiri atas
+	Dan
()	Opsional (bisa ada dan bisa tidak ada)
[]	Memilih salah satu alternatif
{ }	Pengulangan sebanyak n kali
* *	Komentar
@	Identitas atribut kunci
	Pemisah alternatif simbol []

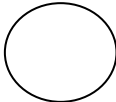

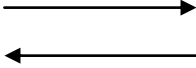
II. 9. Desain Model Aplikasi

Desain model dari aplikasi terdiri dari *physical model* dan *logical model*. *Physical model* dapat digambarkan dengan bagan alir sistem. *Logical model* dalam sistem informasi lebih menjelaskan kepada pengguna bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja. *Logical model* dapat digambarkan dengan DFD (*Data Flow Diagram*) dan kamus data (*Data Dictionary*). Adapun penjelasan dari alat bantu dalam desain model adalah sebagai berikut :

1. Diagram Konteks (*Context Diagram*)

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara proses dan entitas luarnya. Adapun simbol-simbol dalam diagram konteks seperti dijelaskan pada tabel 2.2.


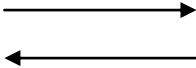
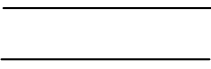

Tabel 2.2 Simbol-Simbol *Context Diagram*

Simbol	Keterangan
	Proses , menunjukkan suatu proses untuk menerima masukan dan menghasilkan keluaran.
	Entitas luar , merupakan sumber atau tujuan dari aliran data dari atau ke sistem. Entitas luar merupakan lingkungan luar sistem
	Arus data atau aliran data , yaitu komponen yang menggambarkan aliran data dari satu proses ke proses lainnya

2. DFD (*Data Flow Diagram*)

DFD merupakan suatu model logika yang menggambarkan asal data dan tujuan data yang keluar dari sistem, serta menggambarkan penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem. Beberapa simbol yang digunakan dalam DFD diterangkan pada tabel 2.3.

Tabel 2.3 Simbol-Simbol DFD

Simbol	Keterangan
	Proses yang berfungsi untuk menunjukkan transformasi dari masukan menjadi keluaran
	Arus data atau aliran data , yaitu komponen yang menggambarkan aliran data dari satu proses ke proses lainnya
	Tempat penyimpanan , yaitu komponen yang digunakan untuk menyimpan kumpulan data, penyimpanan data bisa berupa <i>file</i> , <i>database</i> , maupun bagian dari <i>record</i>
	Entitas luar , merupakan sumber atau tujuan dari aliran data dari atau ke sistem. Entitas luar merupakan lingkungan luar sistem.

II. 10. Perancangan Perangkat Lunak

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya (Sommerville, 2003). Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software* (Pressman, 1997). Tahap desain meliputi perancangan data, perancangan fungsional, dan perancangan antarmuka.

1. Perancangan data

Perancangan data mentransformasikan model data yang dihasilkan oleh proses analisis menjadi struktur data yang dibutuhkan pada saat pembuatan program (*coding*). Selain itu juga akan dilakukan desain terhadap struktur *database* yang akan dipakai.

2. Perancangan fungsional

Perancangan fungsional mendeskripsikan kebutuhan fungsi-fungsi utama perangkat lunak.

3. Perancangan antarmuka

Perancangan antarmuka mendefinisikan bagaimana pengguna (*user*) dan perangkat lunak berkomunikasi dalam menjalankan fungsionalitas perangkat lunak.

II. 11. Implementasi dan Pengujian Unit

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Kemudian pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya (Sommerville, 2003).

Program sebaiknya dirilis setelah dikembangkan, diuji untuk memperbaiki kesalahan yang ditemukan pada pengujian untuk menjamin kualitasnya (Padmini, 2005). Terdapat dua metode pengujian yaitu :

- 1) Metode *white box* yaitu pengujian yang berfokus pada logika internal *software* (*source code* program).
- 2) Metode *black box* yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

Pada tahap pengujian, penulis melakukan metode *black box* yaitu menguji fungsionalitas dari perangkat lunak saja tanpa harus mengetahui struktur internal program (*source code*).

BAB III

ANALISIS DAN PERANCANGAN APLIKASI DATA MINING

Bab ini menjelaskan tentang analisis dan perancangan dalam membangun Aplikasi *Data Mining*. Analisis meliputi analisis *data mining*, analisis lingkungan sistem serta analisis dalam membangun aplikasi.

III. 1. Analisis Data Mining

Dalam penulisan tugas akhir ini akan dicari nilai *support* dan *confidence* dari hubungan tingkat kelulusan dengan data induk mahasiswa. Tidak semua data induk siswa akan dicari hubungannya dengan data kelulusan, hanya beberapa atribut yang kira-kira berguna dan sebarannya tidak terlalu acak. Karena data yang terlalu acak akan membuat proses *mining* memakan waktu lama dan tingkat hubungannya pun rendah. Data induk mahasiswa yang akan dicari hubungannya meliputi proses masuk, asal sekolah, kota asal sekolah, dan program studi. Adapun yang akan diproses *mining* meliputi :

1. Hubungan tingkat kelulusan dengan proses masuk

Hasil dari proses *mining* ini dapat membantu untuk mengetahui sejauh mana tingkat keberhasilan PSSB dan SPMB.

2. Hubungan tingkat kelulusan dengan asal sekolah dan proses masuk

Dari atribut proses masuk dan asal sekolah dicari hubungan tingkat kelulusan dengan asal sekolah yang melalui proses masuk PSSB dengan harapan dapat mengetahui tingkat keberhasilan mahasiswa dengan sekolah tertentu.

3. Hubungan tingkat kelulusan dengan kota asal sekolah

Hubungan tingkat kelulusan dengan asal kota bermanfaat untuk mengetahui daerah-daerah mana yang mempunyai tingkat keberhasilan tinggi ataupun rendah. Diasumsikan bahwa kota asal sekolah merupakan kota tempat asal mahasiswa.

4. Hubungan tingkat kelulusan dengan program studi

Dari atribut program studi dapat diketahui hubungan tingkat kelulusan dan program studi untuk mengetahui tingkat kelulusan program studi.

III. 1. 1. Sumber Data

Data yang digunakan dalam penulisan tugas akhir ini terdiri dari dua sumber data, yaitu data Induk Mahasiswa dan data Kelulusan.

1. Data Induk Mahasiswa

Data induk mahasiswa adalah data mahasiswa yang didata ketika mahasiswa pertama kali masuk perguruan tinggi setelah melakukan registrasi ulang. Data yang dicatat adalah identitas pribadi mahasiswa dan identitas sekolah asal mahasiswa. Proses pendataan dilakukan di tingkat universitas, setelah direkapitulasi kemudian disebarakan ke fakultas masing-masing. Atribut yang ada dapat dilihat dalam tabel 3.1.

Tabel 3.1 Tabel Data Induk Mahasiswa

Atribut	Keterangan
NIM (Nomor Induk Mahasiswa)	Nomor Induk Mahasiswa atau yang disingkat dengan NIM adalah kode yang dimiliki mahasiswa sebagai nomer unik identitas diperguruan tinggi.
Jenis kelamin	Merupakan jenis kelamin mahasiswa yang bersangkutan
Nama mahasiswa	Merupakan nama lengkap mahasiswa yang bersangkutan
Kota lahir	Merupakan kota kabupaten atau kota madya tempat mahasiswa bersangkutan dilahirkan
Tanggal lahir	Merupakan tanggal mahasiswa yang bersangkutan dilahirkan
Agama	Merupakan agama yang dianut mahasiswa yang bersangkutan
Proses masuk	Merupakan jenis jalur masuk ke perguruan tinggi yang diikuti mahasiswa bersangkutan. Proses masuk Universitas Diponegoro dalam rentang tahun 2000-2003 masih menggunakan SPMB (Seleksi Penerimaan Mahasiswa Baru) dan PSSB (Program Seleksi Siswa Berpotensi).

Atribut	Keterangan
Alamat mahasiswa	Merupakan alamat mahasiswa asal yang bersangkutan.
Nama wali	Merupakan nama orang tua atau wali mahasiswa yang bersangkutan.
Alamat wali	Merupakan alamat orang tua atau wali mahasiswa yang bersangkutan
Pendidikan Wali	Merupakan pendidikan orang tua atau wali mahasiswa yang bersangkutan
Nama asal sekolah	Merupakan asal sekolah menengah lanjutan dari mahasiswa yang bersangkutan
Kota asal sekolah	Merupakan kota asal sekolah menengah lanjutan dari mahasiswa yang bersangkutan
Tahun lulus asal sekolah	Merupakan tahun lulus dari asal sekolah menengah lanjutan mahasiswa yang bersangkutan
Status asal sekolah	Merupakan status asal sekolah menengah lanjutan mahasiswa yang bersangkutan.
Jurusan asal sekolah	Merupakan jurusan di asal sekolah menengah lanjutan

2. Data Kelulusan

Data Kelulusan adalah data mahasiswa yang telah dinyatakan lulus. Data yang dicatat adalah identitas mahasiswa dan data kelengkapan kelulusan. Data yang dicatat dapat dilihat pada tabel 3.2

Tabel 3.2 Tabel Data Kelulusan

Atribut	Keterangan
NIM	Nomor Induk Mahasiswa (NIM) adalah kode yang dimiliki mahasiswa sebagai nomer unik identitas diperguruan tinggi. Terdiri dari 9 digit yang merepresentasikan fakultas, jurusan, dan angkatan masuk.
Nama Mahasiswa	Merupakan nama lengkap mahasiswa yang bersangkutan

Atribut	Keterangan
Tempat, Tanggal lahir	Merupakan kota kabupaten atau kotamadya tempat dan tanggal mahasiswa yang bersangkutan dilahirkan
Program Studi	Program studi dari mahasiswa yang bersangkutan
Tanggal lulus	Merupakan tanggal mahasiswa yang bersangkutan dinyatakan lulus
Indeks Prestasi Kumulatif (IPK)	Indeks Prestasi Kumulatif (IPK) adalah ukuran kemampuan mahasiswa sampai pada waktu tertentu yang dapat dihitung berdasarkan jumlah (satuan kredit semester) SKS mata kuliah yang diambil sampai pada periode tertentu dikalikan dengan nilai bobot masing-masing mata kuliah dibagi dengan jumlah seluruh SKS mata kuliah (Anonim, 2009).
Lama studi	Merupakan lama tempuh studi dihitung dimulai saat terdaftar sebagai mahasiswa sampai dinyatakan lulus.
Nama wali	Merupakan nama orang tua atau wali mahasiswa yang bersangkutan.
Alamat wali	Merupakan alamat orang tua atau walai mahasiswa yang bersangkutan
Judul skripsi	Merupakan judul skripsi dari mahasiswa yang bersangkutan
Periode wisuda	Merupakan periode wisuda yang diikuti oleh mahasiswa yang bersangkutan. Di Universitas Diponegoro terdapat 4 periode wisuda.

Data induk mahasiswa yang diambil dalam sampel adalah data mahasiswa angkatan 2000, 2001, 2002 dan 2003. Hal ini didasarkan pada kebutuhan data yang akan di hubungkan dengan data kelulusan, dengan asumsi bahwa mahasiswa

angkatan 2000 -2003 akan lulus dari rentang waktu tahun 2004-2008. Sedangkan data kelulusan yang diambil adalah data kelulusan dari tahun 2004 sampai 2008.

Kedua data tersebut diperoleh dari bagian akademik fakultas MIPA Universitas Diponegoro. Data yang diambil hanya dari mahasiswa sarjana (S1) reguler.

III. 1. 2. Data Yang Digunakan

Dalam penulisan tugas akhir kali ini dicari hubungan beberapa atribut dari data induk mahasiswa dengan tingkat kelulusan. Karena tidak semua tabel digunakan maka perlu dilakukan pembersihan data agar data yang akan diolah benar-benar relevan dengan yang dibutuhkan. Pembersihan ini penting guna meningkatkan performa dalam proses *mining*. Cara pembersihan dengan menghapus atribut yang tidak terpakai dan menghapus data-data yang tidak lengkap isiananya. atribut yang digunakan terdiri dari atribut pada data kelulusan dan pada data induk mahasiswa.

Atribut yang digunakan dalam data induk mahasiswa meliputi :

1. Atribut NIM digunakan sebagai *primary key* untuk menghubungkan dengan data kelulusan
2. Atribut proses masuk digunakan untuk proses *mining* guna mengetahui hubungan antara tingkat kelulusan dengan jalur masuk yang digunakan mahasiswa.
3. Atribut nama asal sekolah digunakan untuk proses *mining* guna mengetahui hubungan antara tingkat kelulusan dengan asal sekolah.
4. Atribut kota asal sekolah digunakan untuk proses *mining* guna mengetahui hubungan tingkat kelulusan dengan kota asal mahasiswa.

Atribut yang digunakan dalam data kelulusan meliputi :

1. NIM digunakan sebagai *primary key* untuk menghubungkan dengan data induk mahasiswa.
2. Indeks Prestasi Kumulatif (IPK) digunakan sebagai ukuran tingkat kelulusan mahasiswa
3. Lama studi digunakan sebagai ukuran tingkat kelulusan mahasiswa.

4. Program studi digunakan untuk proses *mining* guna mengetahui hubungan tingkat kelulusan dengan program studi.

III. 1. 3. Integrasi Data

Dalam penulisan tugas akhir kali ini diasumsikan bahwa data yang diambil sudah berupa tabel-tabel dalam satu *server*. Untuk proses *mining*, data kelulusan dan data induk mahasiswa digabungkan dengan *primary key* NIM. Setelah itu baru dilakukan proses *mining*. Proses integrasi data dilakukan ketika proses ETL (*ekstract, transform, and Load*) ketika membangun *data warehouse*, dalam proses ETL data dalam *data source* digabungkan menjadi satu dalam *data warehouse* dengan key NIM.

III. 1. 4. Transformasi Data

Transformasi data merupakan proses pengubahan atau penggabungan data ke dalam format yang sesuai untuk diproses dalam *data mining*. Seringkali data yang akan digunakan dalam proses *data mining* mempunyai format yang belum langsung bisa digunakan, oleh karena itu perlu dirubah formatnya.

Dalam penulisan tugas akhir ini penulis mencari keterkaitan antara tingkat kelulusan dengan data induk mahasiswa. Tingkat kelulusan mahasiswa dapat dilihat dari lama studi dan IPK (Indeks Prestasi Kumulatif). Dari dua parameter tersebut data diubah menjadi tipe data yang memudahkan untuk diproses. Tingkat kelulusan diukur dari lama studi dan IPK, lama studi dikategorikan berdasarkan peraturan akademik BAB I pasal 1 ayat 2 yang berbunyi “Program sarjana (S1) reguler adalah program pendidikan akademik setelah pendidikan menengah, yang memiliki beban studi sekurang-kurangnya 144 (seratus empat puluh empat) sks dan sebanyak-banyaknya 160 (seratus enam puluh) sks yang dijadwalkan untuk 8 (delapan) semester dan dapat ditempuh dalam waktu kurang dari 8 (delapan) semester dan paling lama 14 (empat belas) semester.” sedangkan IPK dikategorikan berdasarkan predikat kelulusan yang diatur dalam peraturan akademik BAB IV pasal 19 ayat 1 yang berbunyi “predikat kelulusan program sarjana dan program diploma adalah sebagai berikut : ”

Tabel 3.3 Predikat Kelulusan

Indeks Prestasi Kumulatif	Predikat
2,00 – 2,75	Memuaskan
2,76 – 3,50	Sangat memuaskan
3,51 – 4,00	Dengan pujian (cumlaude)

Dari tabel 3.3 data kelulusan berdasarkan IPK dapat dikategorikan menjadi tiga yaitu :

1. IPK memuaskan dengan IPK 2,00 – 2,75
2. IPK sangat memuaskan dengan IPK 2,76 – 3,50
3. IPK tipe dengan pujian dengan IPK 3,51 – 4,00

Pengkategorian data kelulusan berdasarkan lama studi yaitu :

1. Sesuai jadwal, bila lama studi 4 tahun atau kurang dari 4 tahun
2. Tidak sesuai jadwal, bila lama studi lebih dari 4 tahun

Dari dua pengkategorian tersebut dapat dibuat kategori berdasarkan kombinasi keduanya, seperti yang dapat dilihat pada tabel 3.4.

Tabel 3.4 Transformasi Data

Kategori	Keterangan
A1	lama studi 4 tahun atau kurang dari 4 tahun dan IPK 3,51 – 4,00
A2	lama studi 4 tahun atau kurang dari 4 tahun dan IPK 2,76 – 3,50
A3	lama studi 4 tahun atau kurang dari 4 tahun dan IPK 2,00 – 2,75
B1	lama studi lebih dari 4 tahun dan IPK 3,51 – 4,00
B2	lama studi lebih dari 4 tahun dan IPK 2,76 – 3,50
B3	lama studi lebih dari 4 tahun dan IPK 2,00 – 2,75

Dari kombinasi yang terdapat di tabel 3.4 terdapat enam tingkatan untuk mengukur tingkat kelulusan mahasiswa.

III. 1. 5. Penggunaan Algoritma *Apriori*

Algoritma *apriori* adalah algoritma paling terkenal untuk menemukan pola frekuensi tinggi. Pola frekuensi tinggi adalah pola-pola item di dalam suatu *database* yang memiliki frekuensi atau *support* di atas ambang batas tertentu yang disebut dengan istilah minimum *support* atau *threshold*. *Threshold* adalah batas minimum transaksi. Jika jumlah transaksi kurang dari *threshold* maka item atau kombinasi item tidak akan diikuti perhitungan selanjutnya. Penggunaan *threshold* dapat mempercepat perhitungan.

Algoritma *apriori* dibagi menjadi beberapa tahap yang disebut iterasi. Tiap iterasi menghasilkan pola frekuensi tinggi dengan panjang yang sama dimulai dari pass pertama yang menghasilkan pola frekuensi tinggi dengan panjang satu. Di iterasi pertama ini, *support* dari setiap item dihitung dengan men-scan *database*. Setelah *support* dari setiap item didapat, item yang memiliki *support* lebih besar dari minimum *support* dipilih sebagai pola frekuensi tinggi dengan panjang 1 atau sering disingkat 1-itemset. Singkatan k-itemset berarti satu set yang terdiri dari k item.

Iterasi kedua menghasilkan 2-itemset yang tiap set-nya memiliki dua item. Pertama dibuat kandidat 2-itemset dari kombinasi semua 1-itemset. Lalu untuk tiap kandidat 2-itemset ini dihitung *support*-nya dengan men-scan *database*. *Support* artinya jumlah transaksi dalam *database* yang mengandung kedua item dalam kandidat 2-itemset. Setelah *support* dari semua kandidat 2-itemset didapatkan, kandidat 2-itemset yang memenuhi syarat minimum *support* dapat ditetapkan sebagai 2-itemset yang juga merupakan pola frekuensi tinggi dengan panjang 2.

Contoh proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan proses masuk. Misal data seperti pada tabel 3.5

Tabel 3.5 Data Awal

NIM	Kategori Kelulusan	Proses Masuk
J2A003002	A1	PSSB
J2A003003	A2	SPMB
J2A003004	A1	PSSB
J2A003005	A3	SPMB
J2A003006	B2	SPMB
J2A003007	A3	SPMB
J2A003008	A3	SPMB
J2A003009	A2	PSSB
J2A003011	A2	PSSB
J2A003012	A2	PSSB
J2A003013	B2	SPMB

Dari data awal tersebut didapat kandidat pertama (C1) seperti pada tabel 3.6.

Tabel 3.6 Kandidat Pertama (C1)

Itemset	Count
A1	2
A2	4
A3	3
B2	2
PSSB	5
SPMB	6

Ditetapkan $threshold = 3$, maka kandidat yang nilainya kurang dari 3 akan dihapus. Sehingga, didapat hasil seperti pada tabel 3.7 (L1).

Tabel 3.7 hasil setelah $threshold$ ditetapkan (L1)

Itemset	Count
A2	4
A3	3
PSSB	5
SPMB	6

Dari tabel 3.7 didapat kandidat kedua (C2) seperti pada tabel 3.8

Tabel 3.8 Kandidat kedua (C2)

Itemset	Count
A2, PSSB	3
A2, SPMB	1
A3, PSSB	0
A3, SPMB	3

Setelah ditetapkan *threshold* menghasilkan data seperti pada tabel 3.9

Tabel 3.9 Hasil kedua (L2)

Itemset	Count
A2, PSSB	3
A3, SPMB	3

Dari pada tabel 3.9 dapat diambil hasil sebagai berikut :

$$\begin{aligned} \text{Support A2, PSSB} &= \text{Count (A2,PSSB)}/\text{jumlah transaksi} \\ &= 3/11 \end{aligned}$$

$$\begin{aligned} \text{Support A3, SPMB} &= \text{Count(A3, SPMB) } / \text{jumlah transaksi} \\ &= 3/11 \end{aligned}$$

$$\begin{aligned} \text{Confidence A2, PSSB} &= \text{Count(A2,PSSB)}/\text{Count (A2)} \\ &= 3/4 \end{aligned}$$

$$\begin{aligned} \text{Confidence A3, SPMB} &= \text{Count(A3,SPMB)}/\text{Count(A3)} \\ &= 3/3 \end{aligned}$$

Dapat dilihat bahwa proses *mining* hubungan tingkat kelulusan dengan proses masuk mahasiswa dengan *threshold* 3 menghasilkan hubungan A2, PSSB mempunyai nilai *support* = 3/11 *Confidence* = 3/5 dan hubungan A3, SPMB mempunyai nilai *support* = 3/11 *Confidence* = 3/5 mempunyai PSSB mempunyai tingkat kelulusan A2 dan SPMB mempunyai tingkat kelulusan A3 sehingga dapat disimpulkan bahwa mahasiswa yang melalui proses masuk PSSB mempunyai tingkat kelulusan lebih bagus dibanding mahasiswa yang melalui proses masuk SPMB.

Contoh kedua dari proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan proses masuk dan asal sekolah. Misalkan data seperti pada tabel 3.10.

Tabel 3.10 Data awal contoh kedua

NIM	Kategori Kelulusan	Proses masuk	Asal sekolah
J2A003002	A1	PSSB	Semarang
J2A003003	A2	SPMB	Kudus
J2A003004	A1	PSSB	Jepara
J2A003005	A3	SPMB	Semarang
J2A003006	B2	SPMB	Semarang
J2A003007	A3	SPMB	Kudus
J2A003008	A3	SPMB	Jepara
J2A003009	A2	PSSB	Semarang
J2A003011	A2	PSSB	Pati
J2A003012	A2	PSSB	Semarang
J2A003013	B2	SPMB	Kudus

Dari data awal tersebut didapat kandidat pertama (C1) seperti pada tabel 3.11.

Tabel 3.11 Kandidat Pertama (C1) contoh kedua

Itemset	Count
A1	2
A2	4
A3	3
B2	2
PSSB	5
SPMB	6
Semarang	5
Pati	1
Kudus	3
Jepara	2

Ditetapkan $threshold = 3$, maka kandidat yang nilainya kurang dari 3 akan dihapus. Sehingga, didapat hasil seperti pada tabel 3.12 (L1) :

Tabel 3.12 hasil setelah *threshold* ditetapkan (L1) contoh kedua

Itemset	Count
A2	4
A3	3
PSSB	5
SPMB	6
Semarang	5
Kudus	3

Dari tabel 3.12 didapat kandidat kedua (C2) seperti pada tabel 3.13

Tabel 3.13 Kandidat kedua (C2) contoh kedua

Itemset	Count
A2, PSSB	3
A2, SPMB	1
A3, PSSB	0
A3, SPMB	3
A2, semarang	2
A2, Kudus	1
A3, semarang	1
A3, Kudus	1
PSSB, Semarang	3
PSSB, Kudus	0
SPMB, Semarang	2
SPMB, Kudus	3

Setelah ditetapkan *threshold* menghasilkan data seperti pada tabel 3.14

Tabel 3.14 Hasil kedua (L2) contoh kedua

Itemset	Count
A2, PSSB	3
A3, SPMB	3
PSSB, Semarang	3
SPMB, Kudus	3

Dari tabel 3.14 didapat kandidat ketiga (C3) seperti pada tabel 3.15

Tabel 3.15 Kandidat ketiga (C3)

Itemset	Count
A2, PSSB, Semarang	3
A3, SPMB, kudus	3

Dari data-data tersebut dapat diambil hasil sebagai berikut :

$$\begin{aligned} \text{Support (A2, PSSB, Semarang)} &= \text{Count(A2,PSSB,Semarang)}/\text{jumlah transaksi} \\ &= 3/11 \end{aligned}$$

$$\begin{aligned} \text{Support (A3, SPMB, Kudus)} &= \text{Count(A3, SPMB, Kudus) } / \text{jumlah transaksi} \\ &= 3/11 \end{aligned}$$

$$\begin{aligned} \text{Confidence (A2, PSSB, Semarang)} &= \text{Count(A2,PSSB, Semarang)}/\text{Count(A2)} \\ &= 3/4 \end{aligned}$$

$$\begin{aligned} \text{Confidence (A3, SPMB, Kudus)} &= \text{Count(A3,SPMB, Kudus)}/\text{Count(A3)} \\ &= 3/3 \end{aligned}$$

Ditetapkan minimum *support* atau *threshold* adalah 3. Pada iterasi pertama, item yang *support*-nya atau *count*-nya kurang dari 3 dieliminasi dari 1-itemset L1. Kemudian kandidat 2-itemset C2 dari iterasi kedua dibentuk dari *cross product* item-item yang ada di L1. Setelah kandidat 2-itemset itu dihitung dari *database*, ditetapkan 2-itemset L2. Proses serupa berulang di iterasi ketiga, tetapi selain {A2, PSSB, Semarang} dan {A3, SPMB, kudus} yang menjadi kandidat 3-itemset C3 sebenarnya ada juga itemset {A2, PSSB, kudus} dan {A3, SPMB, Semarang} yang dapat diperoleh dari kombinasi item-item di L2, tetapi kedua itemset itu dipangkas karena {PSSB, kudus} dan {SPMB, Semarang} tidak ada di L2. Proses ini berulang sampai tidak ada lagi kandidat baru yang dapat dihasilkan dari minimum *threshold*.

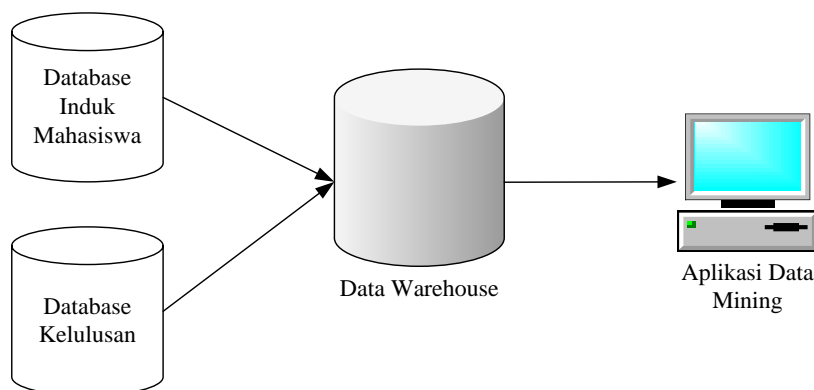
Dalam contoh tabel 3.10 bisa dilihat bahwa algoritma *apriori* dapat mengurangi jumlah kandidat yang harus dihitung *support*-nya dengan pemangkasan. Misalnya kandidat 3-itemset dapat dikurangi dari 3 menjadi 1 saja. Pengurangan jumlah kandidat ini merupakan sebab utama peningkatan performa algoritma *apriori*.

III. 1. 6. Report dan Penyajian Hasil Proses

Setelah proses *mining* akan disajikan hasil dari *data mining* berupa tabel hubungan kekuatan dengan nilai *support* dan *confidence* masing-masing atribut serta *threshold* yang digunakan. Semakin tinggi nilai *confidence* dan *support* maka semakin kuat nilai hubungan antar atribut.

III. 2. Analisis Lingkungan Sistem

Dalam pembangunan Aplikasi *Data Mining* ini, sumber data diperoleh dari dua *database* terpisah yang tidak saling terkait satu sama lain. Karena data mining membutuhkan data dalam jumlah besar, untuk itu diperlukan suatu *data warehouse* yang dapat menampung dan menyatukan dari kedua sumber data tersebut. Selain itu penggunaan *data warehouse* juga bertujuan agar data transaksional dalam kedua *database* sumber tidak terganggu. Ilustrasi aliran data dapat dilihat pada gambar 3.1.



Gambar 3.1 Aliran data dalam proses *data mining*

III. 3. Analisis Perangkat Lunak

III. 3. 1. Deskripsi Umum Perangkat Lunak

Perangkat Lunak yang dikembangkan dalam Aplikasi *Data Mining* ini berbasis *desktop*, dengan *database* lokal.

Pengguna dalam aplikasi adalah pihak penentu kebijakan dalam analisis tingkat kelulusan mahasiswa sehingga dapat diambil langkah-langkah strategis guna meningkatkan tingkat kelulusan. Untuk menjaga kerahasiaan data, maka pengguna dibatasi hanya kepada pihak yang berwenang menggunakan data induk mahasiswa dan data kelulusan.

Sebagai suatu rangkaian proses, *data mining* dibagi dalam beberapa tahap seperti yang sudah diterangkan pada sub bab 3.1. Begitu juga dalam membangun perangkat lunak diperlukan tahapan-tahapan dari analisis, perancangan sampai aplikasi. Sehingga dalam membangun Aplikasi *Data Mining*, tahapan *data mining* sejalan dengan tahapan dalam membangun perangkat lunak. Analisis dari tahapan *data mining* menjadi acuan dalam analisis dan perancangan Aplikasi *Data Mining* ini.

III. 3. 2. Spesifikasi Kebutuhan Fungsional

Spesifikasi kebutuhan fungsional pada Aplikasi *Data Mining* ini merujuk pada kebutuhan akan perancangan *data mining*, seperti yang tertera berikut ini :

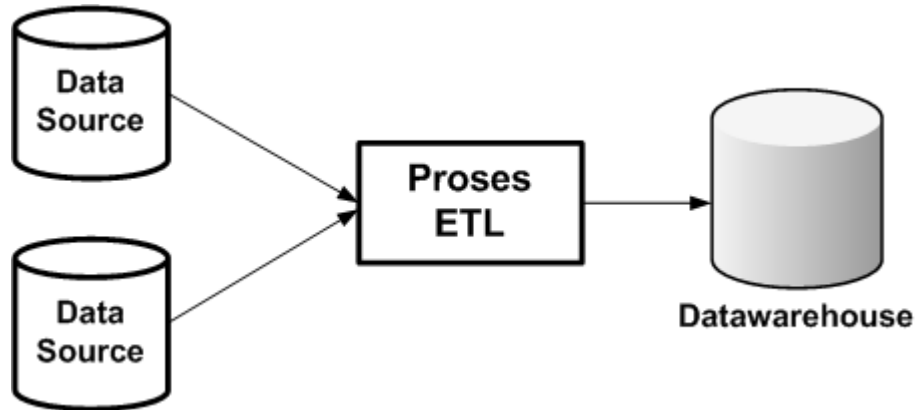
1. Dapat menggabungkan data yang akan diproses *mining* dari data kelulusan dan data induk mahasiswa
2. Dapat menghapus data-data yang tidak relevan serta atribut yang tidak dipakai
3. Dapat merubah data menjadi data yang siap diproses
4. Dapat memproses data untuk *dimining* meliputi :
 - Hubungan tingkat kelulusan dengan proses masuk
 - Hubungan tingkat kelulusan dengan asal sekolah yang melalui proses masuk PSSB
 - Hubungan tingkat kelulusan dengan asal kota
 - Hubungan tingkat kelulusan dengan program studi
5. Dapat menampilkan hasil proses *mining* dengan nilai *support* dan *confidence*

III. 3. 3. Pemodelan Data

Dalam aplikasi ini dibangun *data warehouse* yang digunakan untuk menampung data dari *database* induk mahasiswa dan *database* kelulusan mahasiswa. Tujuan utama pembangunan *data warehouse* adalah agar *database* sumber tidak terganggu bila terjadi *error*, selain itu *data warehouse* memudahkan dalam menyatukan data dari dua *database* sumber.

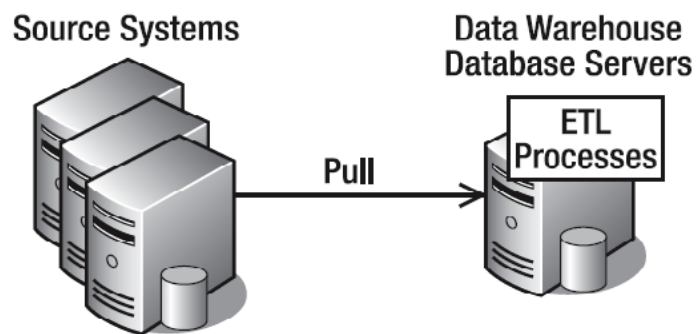
Proses ETL (*Ekstrak Transform Load*) dalam pembangunan *data warehouse* merupakan proses yang penting karena menentukan pembangunan *data warehouse* selanjutnya. Pada pembangunan *data warehouse* ini, terdapat dua *source system*

yaitu dari *database* induk mahasiswa dan *database* kelulusan. Proses pemuatan data dari *data source* ke *data warehouse* melalui proses ETL. Diagram Proses ETL dapat digambarkan seperti pada gambar 3.2



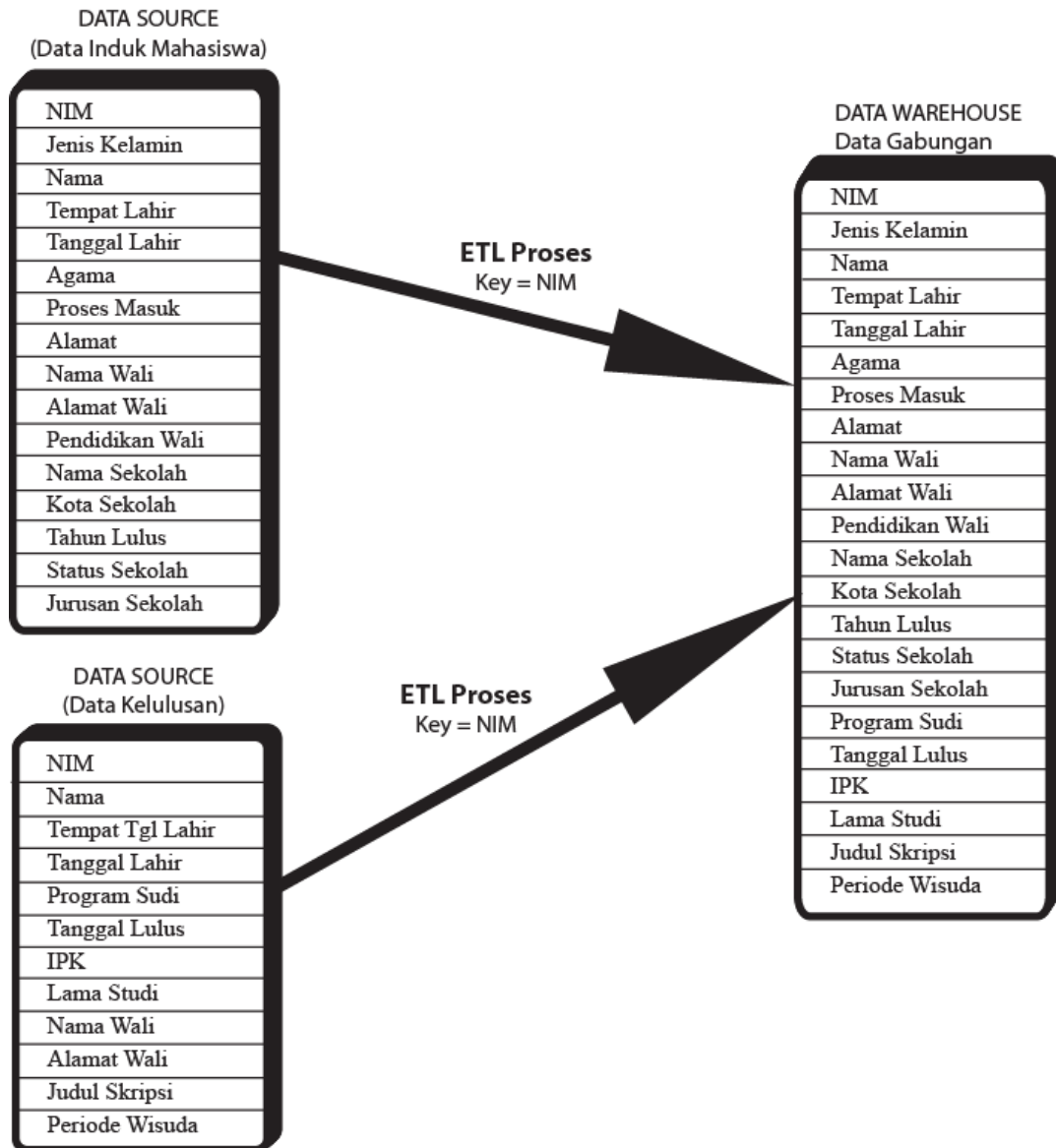
Gambar 3.2 Proses ETL

Dalam pembangunan *data warehouse* ini proses ETL dibangkitkan disisi *storage* tujuan seperti digambarkan pada gambar 3.3. Data pada *source* di *pull* ke *data warehouse* dengan proses ETL yang terletak di *data warehouse*.



Gambar 3.3 Proses Pembangkitan ETL

Database Induk mahasiswa dan *database* kelulusan merupakan *data source* dalam pembangunan *data warehouse*. Dalam proses ETL kedua *data source* tersebut di *load* kedalam *data warehouse*. Proses ETL mencakup proses integrasi data dari data induk dan data kelulusan menjadi data gabungan dengan key NIM. Proses *load* data dan integrasi dapat dilihat dalam gambar 3.4



Gambar 3.4 Proses *load* data

Semua data yang ada pada *data source* diimport ke dalam *data warehouse*, kecuali jika ada data yang sama dari dua *data source* seperti, nama, alamat wali dan yang lain, maka yang di *load* hanya salah satu dari *data source*, dengan asumsi memiliki nilai yang sama. Hal ini dilakukan agar tidak ada *redudancy* data atau data rangkap.

Dalam membangun *data warehouse*, model data yang paling sering digunakan adalah multidimensi dengan bentuk *star schema*, *snowflake schema*, atau *fact constellation schema*. Karena setelah proses ETL hanya terbentuk satu tabel

yang sudah normal dan tidak memiliki dimensi lebih lanjut maka skema hanya berupa satu tabel saja yaitu tabel data gabungan

Dari skema tersebut dapat dibangun *database* baru untuk menampung data-data dari *database* induk mahasiswa dan *database* kelulusan mahasiswa. Karena hanya terdapat satu entitas maka tidak ada relasi antar entitas, sehingga hanya terbentuk satu tabel. Adapun atribut tabel yang terbentuk dapat dilihat pada tabel 3.16.

Tabel 3.16 Tabel atribut data gabungan

Atribut	Keterangan
NIM (Nomor Induk Mahasiswa)	<i>Not null</i> , sebagai key dalam integrasi
Jenis kelamin	-
Nama mahasiswa	-
Kota lahir	-
Tanggal lahir	-
Agama	-
Proses masuk	Indeks
Alamat mahasiswa	-
Nama wali	-
Alamat wali	-
Pendidikan Wali	-
Nama asal sekolah	Indeks
Kota asal sekolah	Indeks
Tahun lulus asal sekolah	-
Status asal sekolah	-
Jurusan asal sekolah	-
Program Studi	Indeks
Indeks Prestasi Kumulatif (IPK)	-
Lama studi	-
Judul skripsi	-
Periode wisuda	-

Pada desain *database*, kamus data digunakan untuk mendefinisikan *file-file* yang ada di dalam *database* dengan lengkap. Hal ini sangat diperlukan untuk membuat *file* secara fisik. Adapun kamus datanya adalah sebagai berikut :

Data Gabungan

Data Induk Mahasiswa =

{ NIM + Jenis Kelamin + Nama Mahasiswa + Kota Lahir + Tanggal lahir + agama + proses masuk + alamat mahasiswa + nama wali + pendidikan wali + nama asal sekolah + Kota asal sekolah + tahun lulus asal sekolah + status asal sekolah + jurusan asal sekolah + Program Studi + Tanggal Lulus + IPK + Lama Studi + Judul Skripsi + Periode Wisuda }

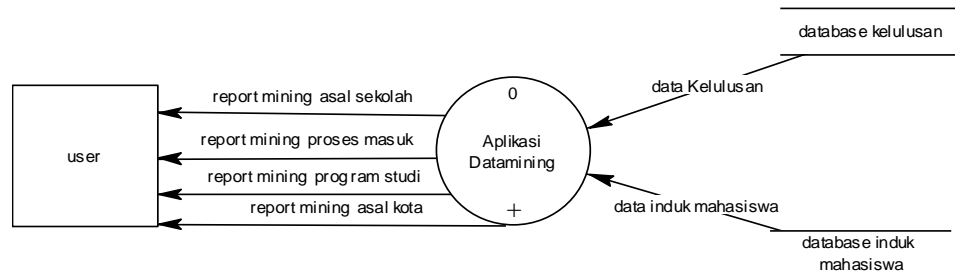
NIM	= {nvarchar}9, <i>Not Null</i>
jenisKelamin	= {nvarchar}9
namaMahasiswa	= {nvarchar}50
tempatLahir	= {nvarchar}25
tanggalLahir	= {date}
agama	= {nvarchar}15
prosesMasuk	= {nvarchar}15
alamatMahasiswa	= {nvarchar}100
namaWali	= {nvarchar}50
pendidikanWali	= {nvarchar}15
namaSekolah	= {nvarchar}50
KotaSekolah	= {nvarchar}30
tahunLulus	= {year}
statusSekolah	= {nvarchar}15
jurusanSekolah	= {nvarchar}15
programStudi	= {nvarchar}20
tanggalLulus	= {date}
IPK	= {float}
lamaStudiThn	= {integer}
lamaStudiBln	= {integer}
judulSkripsi	= {text}
periodeWisuda	= {nvarchar}5

Character = [A-Z | a-z | 0-9]
 Numeric = [0-9]

III. 3. 4. Pemodelan Fungsi

Pemodelan fungsi digambarkan dengan DCD (*Data Context Diagram*), DFD (*Data Flow Diagram*) dan kamus data (*Data Dictionary*).

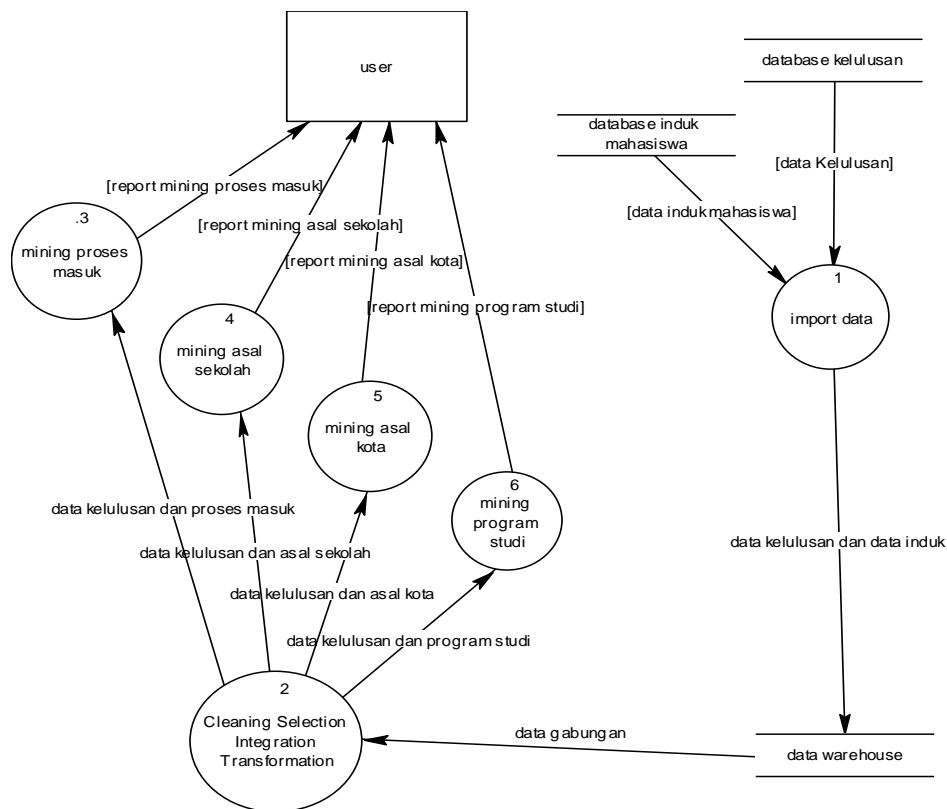
1. DCD / DFD Level-0



Gambar 3.5 DFD Level-0

Gambar 3.5 merupakan DCD / DFD *level-0* pada Aplikasi *Data Mining* yang terdiri dari 2 input dan 4 output. *External entity* berupa pengguna atau *user* dan dua *database* yaitu *database Kelulusan* dan *database Induk Mahasiswa*.

2. DFD Level-1



Gambar 3.6 DFD Level-1

Gambar 3.6 merupakan DFD *Level-1* dari Aplikasi *Data Mining* yang dipecah menjadi beberapa proses kecil guna menjelaskan fungsi-fungsi dan arus data yang mengalir pada Aplikasi *Data Mining*. Berikut proses-proses yang terdapat pada Aplikasi *Data Mining* :

1. *Import Data*

Proses *import* data adalah proses *load* data dari *database* kelulusan dan *database* induk mahasiswa ke *data warehouse*. Semua data akan dimasukkan tanpa ada penyaringan.

2. *Cleaning, Integrasi, Selection, dan transformasi*

- a) *Cleaning* data merupakan proses menghilangkan *noise* dan data yang tidak konsisten atau data yang tidak relevan. Pada umumnya data yang diperoleh, baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Dalam tahap ini semua data yang akan digunakan baik data kelulusan, data induk mahasiswa maupun data nilai semester dibersihkan dari *record* data yang tidak mempunyai atribut lengkap. Selain pembersihan *record* data yang tidak *valid*, juga dilakukan penghapusan atribut yang tidak dipakai, misalnya atribut gaji orang tua, nama orang tua dan lain-lain. Pembersihan data juga akan mempengaruhi performansi dari sistem *data mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.
- b) *Integrasi* data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Data induk mahasiswa, data nilai dan data kelulusan tidak disimpan dalam satu *database*, *Integrasi* data dilakukan pada atribut-atribut yang mengidentifikasi entitas-entitas dengan satu atribut unik yaitu NIM
- c) *Selection data* adalah proses menyeleksi atribut apa yang akan diproses pada mining selanjutnya.
- d) *Transformasi* data merupakan proses mengubah data atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*. Beberapa metode *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Dalam Aplikasi *Data Mining* ini, data yang dirubah yaitu

lama studi dan IPK untuk mengukur tingkat kelulusan. Atribut lama studi dan IPK dibagi menjadi beberapa interval.

3. Proses *mining* proses masuk merupakan proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan proses masuk mahasiswa.
4. Proses *mining* asal sekolah merupakan proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan asal sekolah yang melalui jalur PSSB
5. Proses *mining* asal kota merupakan proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan asal kota mahasiswa, digunakan data kota asal sekolah dengan asumsi kota asal sekolah merupakan kota asal mahasiswa
6. Proses *mining* program studi merupakan proses *mining* untuk mengetahui hubungan tingkat kelulusan dengan program studi.

III. 4. Perancangan Perangkat Lunak

III. 4. 1. Perancangan Fungsi

1. Fungsi Ambil data

Nomor Fungsi : 1

Nama fungsi : Ambil Data

Deskripsi isi : Digunakan untuk mengambil data kelulusan dan data induk mahasiswa kemudian menggabungkannya dengan key NIM dan disimpan dalam tabel data gabungan.

Spesifikasi proses/algoritma :

Initial State (IS) : tabel data gabungan kosong
Final State (FS) : tabel data gabungan terisi
Spesifikasi Proses/algoritma: Ambil data kelulusan dan data induk gabungkan dengan key NIM Simpan dalam tabel data gabungan

2. Fungsi Bersihkan data

Nomor Fungsi : 2

Nama fungsi : Bersihkan data

Deskripsi isi : Digunakan untuk membersihkan data yang tidak sesuai dan tidak lengkap isiannya dari data gabungan dan ditampilkan dalam view data gabungan bersih untuk diolah lebih lanjut. Selain itu penambahan atribut kategori yang berisi transformasi data dari IPK dan lama studi.

Spesifikasi proses/algoritma :

Initial State (IS): view data gabungan bersih kosong
Final State (FS): view data gabungan bersih terisi data gabungan yang telah dibersihkan
Spesifikasi Proses/algoritma: /*pemilihan atribut Ambil data gabungan seleksi atribut yang dipakai dan buang atribut yang tidak lengkap isiannya

3. Fungsi Mining Proses Masuk

Nomor Fungsi : 3

Nama fungsi : Mining Proses Masuk

Deskripsi isi : Digunakan untuk proses *mining* atribut proses masuk

Spesifikasi proses/algoritma :

Initial State (IS): view data gabungan bersih terisi data bersih
Final State (FS): Keluar report hasil proses mining proses masuk
Spesifikasi Proses/algoritma: Hitung masing-masing item dalam Kategori Hitung masing-masing item dalam proses masuk IF jumlah masing-masing item > <i>threshold</i> THEN Hitung kombinasi masing-masing item kategori dan proses masuk Hitung nilai <i>support</i> dan <i>confidence</i> END IF

4. Fungsi Mining Asal Sekolah

Nomor Fungsi : 4

Nama fungsi : Mining Asal Sekolah

Deskripsi isi : Digunakan untuk proses *mining* atribut Asal Sekolah

Spesifikasi proses/algoritma :

Initial State (IS): view data gabungan bersih terisi data bersih
Final State (FS): Keluar report hasil proses mining asal sekolah
Spesifikasi Proses/algoritma: Hitung masing-masing item dalam Kategori Hitung masing-masing item dalam asal sekolah dengan proses masuk PSSB IF jumlah masing - masing item > <i>threshold</i> THEN Hitung kombinasi masing-masing item kategori dan asal sekolah Hitung nilai <i>support</i> dan <i>confidence</i> END IF

5. Fungsi Mining Kota Asal

Nomor Fungsi : 5

Nama fungsi : Mining Kota Asal

Deskripsi isi : Digunakan untuk proses *mining* atribut Kota Asal

Spesifikasi proses/algoritma :

Initial State (IS): view data gabungan bersih terisi data bersih
Final State (FS): Keluar report hasil proses mining kota asal
Spesifikasi Proses/algoritma : Hitung masing-masing item dalam Kategori Hitung masing-masing item dalam Kota Asal IF jumlah masing - masing item > <i>threshold</i> THEN

```

        Hitung kombinasi masing-masing item kategori
dan Kota Asal
        Hitung nilai support dan confidence
END IF

```

6. Fungsi Mining Program Studi

Nomor Fungsi : 6

Nama fungsi : Mining Program Studi

Deskripsi isi : Digunakan untuk proses *mining* atribut Program Studi

Spesifikasi proses/algoritma :

<p>Initial State (IS): view data gabungan bersih terisi data bersih</p>
<p>Final State (FS): Keluar report hasil proses mining program studi</p>
<p>Spesifikasi Proses/algoritma:</p> <pre> Hitung masing -masing item dalam Kategori Hitung masing-masing item dalam Program Studi IF jumlah masing - masing item > <i>threshold</i> THEN Hitung kombinasi masing-masing item kategori dan Program Studi Hitung nilai <i>support</i> dan <i>confidence</i> END IF </pre>

III. 4. 2. Kebutuhan Antarmuka

Pada bagian ini dijelaskan secara rinci semua masukan dan keluaran dari sistem perangkat lunak.

1. Antarmuka Pengguna

Pengguna berinteraksi dengan aplikasi ini dengan menggunakan kakas sebagai berikut :

- *Keyboard*, digunakan oleh pengguna untuk memasukkan data maupun perintah ke dalam aplikasi ini.

- *Mouse*, digunakan untuk melakukan perintah terhadap aplikasi secara modus *Graphical User Interface* (GUI).
- *Monitor*, digunakan oleh pengguna untuk melihat antarmuka dan melihat *report* yang merupakan output dari aplikasi ini.

2. Antarmuka Perangkat Keras

Antarmuka perangkat keras yang digunakan adalah *Personal Computer* dengan *processor single core* dengan kecepatan diatas 2.00 Ghz, RAM 1 GB atau lebih dengan vga 256 bit.

3. Antarmuka Perangkat Lunak

Antarmuka perangkat lunak yang digunakan adalah antarmuka dengan sistem operasi Windows XP/Vista/Seven dengan .Net Framework. Sedangkan perangkat lunak pendukung, seperti :

1. Perangkat lunak pengolah kata
→ Microsoft Office Word 2007
2. Perangkat lunak pengolah project
→ Visual Studio 2008
3. DBMS
→ SQL Server 2005
4. Script Editor
→ Notepad ++
5. Framework
→ .Net Framework

4. Antarmuka Komunikasi

Antarmuka komunikasi diperlukan bila bukan *database* lokal, melainkan harus meremote dari komputer lain, maka yang diperlukan untuk perangkat ini adalah suatu protokol jaringan yang mampu dikenali oleh sistem operasi Windows. Untuk itu diperlukan protokol komunikasi TCP/IP untuk melakukan akses satu komputer dengan komputer lain.

III. 4. 3. Rancangan Tampilan

Dalam Aplikasi *Data Mining* ini terdapat dua buah form. Form pertama merupakan halaman awal yang berisi perintah pengambilan data pemilihan atribut data induk mahasiswa, input *threshold*, perintah proses *mining* dan tombol keluar aplikasi. Form kedua merupakan halaman report *data mining* yang berisi hasil proses *data mining* yaitu tabel nilai *support* dan *confidence*.

Perintah-perintah dalam form pertama berupa tombol ambil data untuk melakukan proses pengambilan data, inputan teks *threshold* untuk memasukkan nilai *threshold*, inputan *combo box* untuk memilih jurusan, inputan *combo box* untuk memilih atribut yang akan diproses *mining*, tombol proses untuk perintah proses *mining* dan tombol keluar untuk perintah keluar aplikasi. Selain tombol tersebut terdapat tabel data hasil dari perintah yang diberikan. Desain tampilan form awal dari Aplikasi *Data Mining* seperti pada gambar 3.7.

Gambar 3.7 Desain Tampilan Form Awal Aplikasi *Data Mining*

Data mining

Hasil Data Mining Tingkat Kelulusan Mahasiswa

Threshold = 0

Jurusan =

Atribut =

Kategori A1

(Lama Studi kurang dari 4 th dan IPK lebih dari 3.5)

Kategori A2

(Lama Studi kurang dari 4 th dan IPK antara 2.76-3.5)

Kategori A3

(Lama Studi kurang dari 4 th dan IPK kurang dari 2.76)

Kategori A1

(Lama Studi lebih dari 4 th dan IPK lebih dari 3.5)

Kategori A2

(Lama Studi lebih dari 4 th dan IPK antara 2.76-3.5)

Kategori A3

(Lama Studi lebih dari 4 th dan IPK kurang dari 2.76)

< kembali

Keluar

Gambar 3.8 Desain tampilan form report Aplikasi *Data Mining*

Hasil keluaran dari proses *data mining* disajikan dalam form report Aplikasi *Data Mining*. Form ini terdiri dari dua informasi utama, yaitu informasi atribut dengan *confidence* tertinggi masing-masing kategori kelulusan dan tabel yang berisi nilai *confidence* dan *support* masing-masing kombinasi tingkat kelulusan dan Atribut. Selain itu terdapat dua tombol perintah yaitu tombol kembali ke menu utama dan tombol untuk keluar aplikasi. Desain tampilan form report Aplikasi *Data Mining* seperti pada gambar 3.8.

BAB IV

IMPLEMENTASI PROGRAM DAN PENGUJIAN

Dalam bab ini menjelaskan tentang implementasi dari perancangan *data mining*, perancangan fungsi, perancangan data dan perancangan tampilan dari Aplikasi *Data Mining*. Selain itu juga terdapat pengujian Aplikasi *Data Mining* dengan pengujian *black box testing*.

IV. 1. Lingkungan Pembangunan

Lingkungan perangkat lunak dan perangkat keras yang digunakan untuk membangun Aplikasi *Data Mining* ini adalah sebagai berikut :

- 1) CPU
 - Prosesor Intel® Celeron® 2.66 Ghz
 - Memori 1024 Gb
 - VGA 128 bit
 - Hardisk 160 GB
- 2) Sistem Operasi : Microsoft® Windows® 7 Professional 6.1
- 3) Editor Script : Notepad ++ v5.0.3.
- 4) Platform
 - Microsoft® Visual Studio® 2008 Version 9.0.21022.8 Professional Edition
 - Microsoft .NET Framework® Version 3.5 SP1
- 5) DBMS :
 - Microsoft® SQL Server® Management Studio 9.00.1399.00
 - Microsoft® MSXML 3.0 4.0 5.0 6.0
 - Microsoft® .NET Framework® 2.0.50727.4927

IV. 2. Implementasi Data

Implementasi rancangan data merupakan transformasi rancangan data yang dihasilkan dari proses perancangan data menjadi suatu *database*. *Database*

merupakan suatu *data warehouse* dengan nama “dataMining” yang dibangun untuk menyimpan data kelulusan mahasiswa dan data induk mahasiswa yang disatukan dengan key NIM dan disimpan dalam tabel data gabungan.

Penjelasan tabel data gabungan beserta *field-field* di dalamnya dapat dilihat pada tabel 4.1.

Tabel 4.1 Struktur tabel data gabungan

Nama <i>Field</i>	Type	Panjang <i>Field</i>	Keterangan
NIM	nvarchar	10	Not Null
jenisKelamin	nvarchar	10	
namaMahasiswa	nvarchar	50	
tempatLahir	nvarchar	25	
tanggalLahir	date		
agama	nvarchar	15	
prosesMasuk	nvarchar	15	
alamatMahasiswa	nvarchar	100	
namaWali	nvarchar	50	
pendidikanWali	nvarchar	15	
namaSekolah	nvarchar	50	
KotaSekolah	nvarchar	30	
tahunLulus	year		
statusSekolah	nvarchar	15	
jurusanSekolah	nvarchar	15	
programStudi	nvarchar	20	
tanggalLulus	Date		
IPK	Float		
lamaStudiThn	Integer		
lamaStudiBln	Integer		
judulSkripsi	Text		
periodeWisuda	integer		

IV. 3. Implementasi Fungsi

Implementasi rancangan fungsi merupakan hasil transformasi dari proses perancangan fungsi yang telah dijelaskan pada sub bab 3.4.1 menjadi modul-modul dalam aplikasi. Hasil dari algoritma tersebut dituangkan dalam list program yang dapat dilihat pada lampiran.

IV. 4. Implementasi Rancangan Antarmuka

Hasil rancangan antarmuka Aplikasi *Data Mining* diimplementasikan dalam dua form. Form pertama merupakan halaman awal yang berisi perintah pengambilan data pemilihan atribut data induk mahasiswa, input *threshold*, perintah proses *mining* dan tombol keluar aplikasi. Sedangkan form kedua berupa informasi hasil proses *data mining* yaitu tabel nilai *support* dan *confidence*.

Tombol dan inputan dalam form awal disusun secara berurutan dan hanya bisa diakses secara terurut. Karena dalam *data mining* proses tersebut bersifat sekuensial. Misalkan pengguna menekan tombol bersihkan data sebelum data diambil maka tombol tersebut tidak akan aktif. Dalam inputan *threshold* jika pengguna menginputkan bukan angka otomatis *threshold* bernilai *default* yaitu nol. Tampilan dari form awal dapat dilihat pada gambar 4.1.

NIM	Nama	Jenis Kelamin	Tempat Lahir	Tanggal Lahir	Agama	Proses Masuk	Alamat
J2A000001	ADI SAYONO	PRIA	JAKARTA TIMUR	1982-03-24T00:0...	ISLAM	PSSB	ADIPURA 2 NO 45 PURW...
J2A000002	ADI WIBOWO	PRIA	PURBALINGGA	1982-03-09T00:0...	ISLAM	PSSB	BAJONG RT3 RW 1 BKT P...
J2A000004	AGUNG NUGROHO	PRIA	PASAMAN	1981-10-14T00:0...	ISLAM	UMPTN	GAWANAN RT 1 RW 2 CL...
J2A000005	ALFONSUS ARIS...	PRIA	PURWOREJO	1981-05-13T00:0...	KATHOLIK	UMPTN	KDGB KALIGESING PUR...
J2A000006	ANAS WAHYUDIN	PRIA	SEMARANG	1981-03-25T00:0...	ISLAM	UMPTN	JL TAMBAK MULYO RT 4S...
J2A000007	ANDY B FITRIANTO	PRIA	BANYUMAS	1982-05-07T00:0...	ISLAM	UMPTN	KR PETIR TAMBAK BANY...
J2A000008	ANGGUN SWASTI...	WANITA	SEMARANG	1982-10-13T00:0...	KATHOLIK	PSSB	JL MINTOJIWO DALAM V ...

Gambar 4.1 Tampilan form awal Aplikasi *Data Mining*

Form kedua yaitu form report *data mining*. Form ini akan aktif dan tampil jika pengguna menekan tombol proses, dalam form ini terdapat hasil dari proses *mining* berupa nilai masing-masing kategori yang mempunyai nilai *confidence* tertinggi. Selain itu terdapat tabel itemset dengan atribut itemset, cacah jumlah itemset atau *count*, *support*, dan *confidence* dari itemset tersebut. Dalam form ini terdapat dua tombol yaitu tombol kembali ke form awal dan tombol keluar. Tampilan pada form report data mining dapat dilihat pada gambar 4.2

Hasil Data Mining Tingkat Kelulusan Mahasiswa

Atribut = Proses Masuk
Threshold = 0
Jurusan = Semua Jurusan

Kategori A1 (Lama studi kurang dari 4 tahun dan IPK lebih dari 3.5) :
Memiliki Hubungan dengan A1,UMPTN dengan nilai kepercayaan = 55,5556%
Hal ini didukung oleh 1,0846% dari seluruh data merupakan kombinasi keduanya.

Kategori A2 (Lama studi kurang dari 4 tahun dan IPK antara 2.76-3.5) :
Memiliki Hubungan dengan A2,UMPTN dengan nilai kepercayaan = 69,2308%
Hal ini didukung oleh 3,9046% dari seluruh data merupakan kombinasi keduanya.

Kategori A3 (Lama studi kurang dari 4 tahun dan IPK kurang dari 2.75) :
Memiliki Hubungan dengan A3,UMPTN dengan nilai kepercayaan = 100%
Hal ini didukung oleh 0,2169% dari seluruh data merupakan kombinasi keduanya.

Kategori B1 (Lama studi lebih dari 4 tahun dan IPK lebih dari 3.5) :
Memiliki Hubungan dengan B1,UMPTN dengan nilai kepercayaan = 56,5217%
Hal ini didukung oleh 2,82% dari seluruh data merupakan kombinasi keduanya.

Kategori B2 (Lama studi Lebih dari 4 tahun dan IPK antara 2.76-3.5) :
Memiliki Hubungan dengan B2,UMPTN dengan nilai kepercayaan = 72,9508%
Hal ini didukung oleh 57,9176% dari seluruh data merupakan kombinasi keduanya.

Kategori B3 (Lama studi lebih dari 4 tahun dan IPK kurang dari 2.75) :
Memiliki Hubungan dengan B3,UMPTN dengan nilai kepercayaan = 72,2222%
Hal ini didukung oleh 5,6399% dari seluruh data merupakan kombinasi keduanya.

Item Set	Count	Confidence (%)	Support (%)
A1,PSSB	3	33,3333	0,6508
A1,TEST LOKAL	1	11,1111	0,2169
A1,UMPTN	5	55,5556	1,0846
A2,PSSB	8	30,7692	1,7354
A2,UMPTN	18	69,2308	3,9046
A3,UMPTN	1	100	0,2169

Kembali Menu Utama Keluar

Hasil Proses Data mining, dengan jurusan = Semua Jurusan, item = Proses Masuk

Gambar 4.2 Tampilan form report *data mining*.

Dalam menggunakan Aplikasi *Data Mining* ini pertama kali pengguna akan masuk dalam halaman awal. Pada halaman awal pengguna melakukan perintah secara sekuensial dari atas ke bawah. Tombol ambil data adalah tombol untuk mengeksekusi fungsi ambil data, yaitu mengambil data kelulusan dan data induk mahasiswa kemudian menyimpannya dalam tabel baru yaitu data gabungan. Pengguna dapat memilih jurusan dengan memilih pada *combo box* pilihan jurusan. Setelah itu pengguna menentukan *threshold* dengan mengisi label teks. Setelah mengisi *threshold* pengguna memilih item yang akan diproses *mining* dengan memilih pada *combo box*. Kemudian pengguna membersihkan data dengan menekan tombol bersihkan data. Sampai pada proses ini data siap diproses *mining*.

Selanjutnya pengguna menekan tombol proses *mining* untuk melakukan proses *mining*. Barulah hasil dari proses *mining* akan keluar pada form ke dua.

Dalam form kedua pengguna dapat melihat dua tampilan hasil data *mining* dan dua tombol. Tampilan pertama merupakan hasil *mining* masing-masing tingkat kelulusan dengan atribut yang paling tinggi nilai *confidence* serta nilainya. Sedangkan tampilan kedua merupakan tabel hasil *mining* seluruh item kategori kelulusan dengan semua item data induk dan nilai *support* dan *confidence*. Tombol dalam tampilan report Aplikasi *Data Mining* merupakan tombol untuk kembali ke form awal dan keluar dari aplikasi.

IV. 5. Pengujian Aplikasi *Data Mining*

IV. 5. 1. Lingkungan Pengujian

Lingkungan perangkat lunak dan perangkat keras yang digunakan untuk membangun Aplikasi *Data Mining* ini meliputi perangkat keras dan perangkat lunak. Spesifikasi perangkat lunak sebagai berikut :

1. Processor : Intel Celeron 2.66
2. RAM : 1024 Mb
3. VGA : 128 bit, 256 Mb
4. Harddisk : 160Gb

Sedangkan spesifikasi perangkat lunaknya sebagai berikut :

1. Sistem Operasi : Microsoft® Windows® 7 Professional 6.1
2. FrameWork : Microsoft .NET Framework® Version 3.5 SP1
3. DBMS : Microsoft® SQL Server® Management Studio
9.00.1399.00
4. XML sistem : Microsoft® MSXML 3.0 4.0 5.0 6.0

IV. 5. 2. Rancangan Pengujian

Dalam pengujian Aplikasi *Data Mining* ini digunakan teknik pengujian *Black Box*. Teknik yang digunakan dalam pengujian *Black Box* antara lain :

1. Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang.
2. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut dan bagaimana hasil dari proses *mining*.
3. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya.

Identifikasi dan pelaksanaan pengujian dapat dilihat pada tabel 4.2.

Tabel 4.2 Identifikasi dan pelaksanaan pengujian

No	Kelas Uji	Butir Uji	Tingkat Pengujian	Jenis Pengujian
1.	Fungsi pengambilan data	Menekan tombol ambil data	Pengujian Sistem	<i>Black Box</i>
2.	Fungsi bersihkan data	Menekan tombol bersihkan data	Pengujian Sistem	<i>Black Box</i>
3.	Fungsi Proses Mining	Menekan tombol proses mining	Pengujian Sistem	<i>Black Box</i>

IV. 5. 3. Hasil Uji

Hasil uji dianggap sukses jika pada tabel pengujian, hasil yang didapat sesuai dengan kriteria evaluasi hasil dan hasil yang diharapkan. Tabel hasil pengujian dapat dilihat pada tabel 4.3

Tabel 4.3 Hasil Uji Aplikasi *Data Mining*

Deskripsi	Prosedur Pengujian	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Simpulan
Fungsi pengambilan data	Menekan tombol ambil data	Tampilan data gabungan pada tabel	Data gabungan muncul pada <i>gridview</i> gabungan	Data gabungan muncul pada <i>gridview</i> gabungan	Diterima
Fungsi bersihkan data	Menekan tombol bersihkan data	Tampilan data gabungan yang sudah bersih dari data kotor dan atribut tidak dipakai	Data gabungan yang sudah bersih muncul pada <i>gridview</i> gabungan bersih	Data gabungan yang sudah bersih muncul pada <i>gridview</i> gabungan bersih	Diterima
Fungsi Proses <i>Mining</i>	Menekan tombol proses <i>mining</i>	Tampilkan form report <i>mining</i> berupa tabel hasil proses mining dengan nilai support dan confidence dan menampilkan masing-masing kategori dengan nilai <i>confidence</i> paling tinggi	Tampil form report dengan hasil report proses <i>mining</i>	Form report dengan tabel hasil proses <i>mining</i> dan tampilan masing-masing kategori dengan nilai <i>confidence</i> paling tinggi	Diterima

IV. 5. 4. Analisis Hasil Uji

Berdasarkan pengujian di atas, dapat disimpulkan bahwa Aplikasi *Data Mining* dapat digunakan untuk mengetahui hubungan data kelulusan dengan data induk mahasiswa.

BAB V

PENUTUP

V. 1. Kesimpulan

Kesimpulan yang dapat diambil dalam pengerjaan tugas akhir ini adalah Aplikasi *Data Mining* ini dapat digunakan untuk menampilkan informasi tingkat kelulusan. Informasi yang ditampilkan berupa nilai *support* dan *confidence* hubungan antara tingkat kelulusan dengan data induk mahasiswa. Semakin tinggi nilai *confidence* dan *support* maka semakin kuat nilai hubungan antar atribut. Data induk mahasiswa yang diproses *mining* meliputi data proses masuk, data asal sekolah, data kota mahasiswa, dan data program studi.

Hasil dari proses *data mining* ini dapat digunakan sebagai pertimbangan dalam mengambil keputusan lebih lanjut tentang faktor yang mempengaruhi tingkat kelulusan khususnya faktor dalam data induk mahasiswa.

V. 2. Saran

Untuk pengembangan Aplikasi *Data Mining* lebih lanjut, dapat menggunakan algoritma lain, misal algoritma *FP-Growth*. Perbedaannya adalah algoritma *apriori* harus melakukan *scan database* setiap kali iterasi, sedangkan algoritma *FP-Growth* hanya melakukan satu kali *scan database* diawal.

DAFTAR PUSTAKA

- [1] Anonim, 2009 “*Peraturan Akademik Universitas Diponegoro Bidang Pendidikan*”, Semarang.
- [2] Bramer, Max, 2007, “*Principles of Data Mining*”, Springer, London.
- [3] Chintakayala, Padmini. 2005. “*Beginners Guide for Software Testing : Symbiosys Technologies*”.
- [4] Davies, and Paul Beynon, 2004, “*Database Systems Third Edition*”, Palgrave Macmillan, New York.
- [5] Elmasri, Ramez and Shamkant B. Navathe, 2000, “*Fundamentals of Database Systems. Third Edition*”, Addison Wesley Publishing Company, New York.
- [6] Han, J. and Kamber, M, 2006, “*Data Mining Concepts and Techniques Second Edition*”. Morgan Kauffman, San Francisco.
- [7] Kadir, Abdul, 1999, “*Konsep dan Tuntunan Praktis Basis Data*”, Penerbit Andi, Yogyakarta.
- [8] Kusrini, dan Emha Taufik Luthfi, 2009, “*Algoritma Data Mining*”, Penerbit Andi, Yogyakarta.
- [9] Pramudiono, I. 2007. *Pengantar Data Mining : Menambang Permata Pengetahuan di Gunung Data*. <http://www.ilmukomputer.org/wp-content/uploads/2006/08/iko-datamining.zip> Diakses pada tanggal 15 Maret 2009 jam 08.54
- [10] Pramudiono, I., 2007, *Algoritma Apriori*, <http://datamining.japati.net/cgi-bin/indodm.cgi?bacaarsip&1172210143> Diakses pada tanggal 25 April 2009 jam 10.00
- [11] Rainardi, Vincent, 2008, “*Building a Data Warehouse with Examples in SQL Server*”, Springer, New York.
- [12] Pressman, Roger S, 1997, “*Software Engineering: A Practitioner’s Approach.*” The McGraw-Hill Companies, Inc., New York
- [13] Santosa, Budi, 2007, “*Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*”, Graha Ilmu, Yogyakarta.
- [14] Sommerville, Ian, 2003, “*Software Engineering (Rekayasa Perangkat Lunak)/ Edisi 6/Jilid 1*” Erlangga, Jakarta.

- [15] Witten, I. H and Frank, E. 2005. *Data Mining : Practical Machine Learning Tools and Techniques Second Edition*. Morgan Kauffman : San Francisco.

Lampiran 1. Code Program

```
'Hal awal
Imports System.Data
Imports System.Data.SqlClient
Public Class formAwal
    'tres merupakan treshold
    Dim tres As Integer = 0
    Dim jmlRecordDataBersih As Integer
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        tolbar.Text = "Ready, Klik tombol Ambil data untuk mengambil data
kelulusan dan data induk"
        'TODO: This line of code loads data into the
'DataSetGabungan.viewGabungan' table. You can move, or remove it, as
needed.
        Me.TableAdapterManager.Fill(Me.DataSetGabungan.viewGabungan)
        DataGridViewDataGabungan.Visible = True
        DataGridViewAsalkota.Visible = False
        DataGridViewAsalSekolah.Visible = False
        DataGridViewProdi.Visible = False
        DataGridViewProsesmsk.Visible = False
        cmbBoxDm.Enabled = False
        BtnBersihkandata.Enabled = False
        btnProses.Enabled = False
        domtres.Enabled = False
        CmbBoxPilJur.Enabled = False
        lblAmbilData.Text = "Impor data dari data induk dan data
kelulusan simpan di data gabungan"
        ProgressBar1.Visible = False
    End Sub
    Private Sub BtnAmbilData_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnAmbilData.Click
        tolbar.Visible = False
        ProgressBar1.Visible = True
        ProgressBar1.Maximum = 50

        Me.DataGabunganTableAdapter.Filltruncate(Me.DataSetGabungan.dataGabungan)
        DataGridViewDataGabungan.Visible = True
        DataGridViewAsalkota.Visible = False
        DataGridViewAsalSekolah.Visible = False
        DataGridViewProdi.Visible = False
        DataGridViewProsesmsk.Visible = False
        CmbBoxPilJur.Focus()
        cmbBoxDm.Enabled = False
        CmbBoxPilJur.Enabled = True
        domtres.Enabled = False
        BtnBersihkandata.Enabled = False
        btnProses.Enabled = False
        Me.TableAdapterManager.Fill(Me.DataSetGabungan.viewGabungan)

        Me.DataSetGabungan.viewGabungan.WriteXml("C:\DataMining\viewGabungan.xml"
)
        ProgressBar1.Value = 10
        Dim objBL As Object
        'membuat objek xml
        objBL = CreateObject("SQLXMLBulkLoad.SQLXMLBulkLoad")
        'memuat koneksi database
        objBL.ConnectionString = "provider=SQLOLEDB;data
source=NUXONGEAR;database=dataMining;integrated security=SSPI;"
        'memanggil dan menjalankan objek xml dengan modul bulk load.
```

```

        objBL.ErrorLogFile = "C:\DataMining\error.log"
        ProgressBar1.Value = 20
        'memanggil file xml dan skema
        objBL.Execute("C:\DataMining\dataGabunganMapping.xml",
"C:\DataMining\viewGabungan.xml")
        objBL = Nothing
        ProgressBar1.Value = 45
        'TODO: This line of code loads data into the
'DataSetGabungan.dataGabungan' table. You can move, or remove it, as
needed.
        Me.DataGabunganTableAdapter.Fill(Me.DataSetGabungan.dataGabungan)
        lblAmbilData.Text = "Proses Load data Sukses, Backup data di
C:\DataMining\viewGabungan.xml"
        ProgressBar1.Value = 50
        ProgressBar1.Visible = False
        tolbar.Visible = True
        tolbar.Text = "Proses Ambil Data selesai, Pilih Jurusan yang akan
di proses"
    End Sub
    Private Sub cmbBoxDm_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbBoxDm.SelectedIndexChanged
        tolbar.Text = "Atribut yang akan di proses adalah " +
cmbBoxDm.SelectedItem + ", klik tombol bersihkan datas untuk membersihkan
data"
        CmbBoxPilJur.Enabled = False
        BtnBersihkandata.Enabled = True
        BtnBersihkandata.Focus()
        lblAtribut.Text = "Atribut yang akan di proses adalah " +
cmbBoxDm.SelectedItem
    End Sub
    Private Sub BtnBersihkandata_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnBersihkandata.Click
        ProgressBar1.Visible = True
        domtres.Enabled = True
        Dim pil As String
        pil = cmbBoxDm.SelectedItem

        'membuat tabel data bersih
        Me.DataBersihTableAdapter.truncate(Me.DataSetGabungan.dataBersih)
        Dim obj As Object
        'membuat objek xml
        obj = CreateObject("SQLXMLBulkLoad.SQLXMLBulkLoad")
        'membuat koneksi database

        obj.ConnectionString = "provider=SQLOLEDB;data
source=NUXONGEAR;database=dataMining;integrated security=SSPI;"
        'memanggil and menjalankan objek xml dengan modul bulk load.
        obj.ErrorLogFile = "C:\DataMining\error.log"

        'memanggil file xml dan skema
        obj.Execute("C:\DataMining\dataBersihMapping.xml",
"C:\DataMining\viewGabungan.xml")

        obj = Nothing
        'TODO: This line of code loads data into the
'DataSetGabungan.dataBersih' table. You can move, or remove it, as
needed.

```

```

Me.DataBersihTableAdapter.Fill(Me.DataSetGabungan.dataBersih)
cmbBoxDm.Enabled = False
Me.BtnBersihkandata.Enabled = False
Me.domtres.Enabled = False

'pembersihan data
Dim pilJur As String
pilJur = CmbBoxPilJur.SelectedItem
If pilJur = "Matematika" Then
    Me.DataBersihTableAdapter.DeleteMat()
ElseIf pilJur = "Statistika" Then
    Me.DataBersihTableAdapter.DeleteStat()
ElseIf pilJur = "Biologi" Then
    Me.DataBersihTableAdapter.DeleteBio()
ElseIf pilJur = "Fisika" Then
    Me.DataBersihTableAdapter.DeleteFisika()
ElseIf pilJur = "Kimia" Then
    Me.DataBersihTableAdapter.DeleteKimia()
ElseIf pilJur = "Teknik Informatika" Then
    Me.DataBersihTableAdapter.DeleteTI()
End If

'menghilangkan noise pada asal sekolah
If pil = "Asal Sekolah" Then
    Me.DataBersihTableAdapter.DeleteAsalSekolah()
End If
'me load lagi tabel data bersih
Me.DataBersihTableAdapter.Fill(Me.DataSetGabungan.dataBersih)
'transformasi data

Dim IPK As Double
Dim studiBln As Integer
Dim studiThn As Integer
Dim kategori As String = ""
Dim NIM As String
jmlRecordDataBersih = Me.DataSetGabungan.dataBersih.Rows.Count()
ProgressBar1.Maximum = jmlRecordDataBersih
domtres.Maximum = jmlRecordDataBersih * 10
domtres.Minimum = 0
For i As Integer = 0 To (jmlRecordDataBersih - 1)
    NIM = Me.DataBersihTableAdapter.GetData(i).Item("NIM")
    IPK = Me.DataBersihTableAdapter.GetData(i).Item("IPK")
    studiThn =
Me.DataBersihTableAdapter.GetData(i).Item("LamaStudiThn")
    studiBln =
Me.DataBersihTableAdapter.GetData(i).Item("LamaStudiBln")
    ProgressBar1.Value = i

    If ((studiThn = 4 And studiBln = 0) Or studiThn < 4) And 3.5
< IPK Then
        kategori = "A1"
    ElseIf ((studiThn = 4 And studiBln = 0) Or studiThn < 4) And
2.75 < IPK And IPK < 3.51 Then
        kategori = "A2"
    ElseIf ((studiThn = 4 And studiBln = 0) Or studiThn < 4) And
0 < IPK And IPK < 2.76 Then
        kategori = "A3"
    ElseIf (studiThn > 4 Or (studiThn = 4 And studiBln > 0)) And
3.5 < IPK Then
        kategori = "B1"

```

```

        ElseIf (studiThn > 4 Or (studiThn = 4 And studiBln > 0)) And
2.75 < IPK And IPK < 3.51 Then
            kategori = "B2"
        ElseIf (studiThn > 4 Or (studiThn = 4 And studiBln > 0)) And
0 < IPK And IPK < 2.76 Then
            kategori = "B3"
        Else
            kategori = "UNDEFINED"
        End If

        Me.DataBersihTableAdapter.updateKategori(kategori, NIM)
    Next i

    'menampilkan data yang sudah bersih ke layar
    If pil = "Proses Masuk" Then
        DataGridViewProsesmsk.Visible = True
        DataGridViewDataGabungan.Visible = False
        DataGridViewAsalkota.Visible = False
        DataGridViewAsalSekolah.Visible = False
        DataGridViewProdi.Visible = False
    ElseIf pil = "Asal Sekolah" Then
        DataGridViewProsesmsk.Visible = False
        DataGridViewDataGabungan.Visible = False
        DataGridViewAsalkota.Visible = False
        DataGridViewAsalSekolah.Visible = True
        DataGridViewProdi.Visible = False
    ElseIf pil = "Kota Asal" Then
        DataGridViewProsesmsk.Visible = False
        DataGridViewDataGabungan.Visible = False
        DataGridViewAsalkota.Visible = True
        DataGridViewAsalSekolah.Visible = False
        DataGridViewProdi.Visible = False
    ElseIf pil = "Jurusan" Then
        DataGridViewProsesmsk.Visible = False
        DataGridViewDataGabungan.Visible = False
        DataGridViewAsalkota.Visible = False
        DataGridViewAsalSekolah.Visible = False
        DataGridViewProdi.Visible = True
    End If

    'TODO: This line of code loads data into the
'DataSetGabungan.dataBersih' table. You can move, or remove it, as
needed.
    Me.DataBersihTableAdapter.Fill(Me.DataSetGabungan.dataBersih)
    'membuat domain threshold
    ProgressBar1.Visible = False
    btnProses.Enabled = True
    domtres.Enabled = True
    cmbBoxDm.Enabled = False
    CmbBoxPilJur.Enabled = False
    domtres.Focus()
    lblBershData.Text = "Pembersihan data selesai. total Record = " &
jmlRecordDataBersih
    tolbar.Visible = True
    tolbar.Text = "proses pembersihan data selesai, masukkan
thershold."
End Sub

Private Sub btnProses_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnProses.Click
    tres = domtres.Value
    Dim pil As String

```

```

Dim frmreport As New FormReport

frmreport.ItemSetTableAdapter.truncate(frmreport.DataSetGabungan.itemSet)
pil = cmbBoxDm.SelectedItem
If pil = "Asal Sekolah" Then
    frmreport.labelJudul2.Text = "Asal Sekolah"
Else
    frmreport.labelJudul2.Text = pil
End If

frmreport.lblTresNilai.Text = tres
btnProses.Enabled = False
Dim sup As Double
Dim conf As Double
Dim antecedent As Integer
'MEMBUAT C1 KATEGORI
Dim countC1Kat As Integer
'menghitung jumlah item C1 dalam kategori
'countC1Kat = jumlah item C1 dalam kategori
countC1Kat =
Me.DataBersihTableAdapter.getItemKategori.Rows.Count()

'menghitung jumlah baris masing2 item C1 dalam kategori
'mengambil nama masing2 item C1 dalam kategori
Dim countC1ItemKat(countC1Kat) As Integer
'countC1ItemKat = jumlah row masing2 item C1 dalam kategori
Dim strC1ItemKat(countC1Kat) As String
'strC1ItemKat = nama masing-masing item C1 dalam kategori
For i As Integer = 1 To countC1Kat
    strC1ItemKat(i) = Me.DataBersihTableAdapter.getItemKategori(i
- 1).Item("kategori")
    countC1ItemKat(i) =
Me.DataBersihTableAdapter.countKategori(strC1ItemKat(i))
Next i
'Didapat nama masing2 item dalam C1 disimpan di array
strC1ItemKat(countC1kat)
'Didapat jumlah baris masing2 item dalam C1 disimpan di array
countC1ItemKat(countC1kat)

'MEMBUAT L1 KATEGORI
'mencari jumlah item kategori dalam C1 yang lebih dari tershold
sebagai batas array L1 (jumlah L1)
Dim maxL1Kat As Integer = 0
'maxL1Kat = jumlah item L1 dalam kategori
For i As Integer = 1 To countC1Kat
    If countC1ItemKat(i) > tres Then
        maxL1Kat = maxL1Kat + 1
    End If
Next i
'maxL1Kat = jumlah L1 dalam kategori

Dim countL1ItemKat(maxL1Kat) As Integer
'countL1ItemKat = jumlah baris masing2 item L1 dalam kategori
Dim strL1ItemKat(maxL1Kat) As String
'strL1ItemKat = nama masing2 item L1 dalam kategori
Dim indexL1Kat As Integer = 1
'indexL1Kat = index baru array L1 dalam kategori
For i As Integer = 1 To countC1Kat
    If countC1ItemKat(i) > tres Then

```

```

        countL1ItemKat(indexL1Kat) = countC1ItemKat(i)
        strL1ItemKat(indexL1Kat) = strC1ItemKat(i)
        indexL1Kat = indexL1Kat + 1
    End If
Next i
'Didapat nama masing2 item dalam L1 disimpan di array
strL1ItemKat(maxL1kat)
'Didapat jumlah baris masing2 item dalam L1 disimpan di array
countL1ItemKat(maxL1kat)

If pil = "Proses Masuk" Then

    'MEMBUAT C1 PROSES MASUK
    'menghitung jumlah item c1 dalam proses masuk
    Dim countC1PM As Integer
    countC1PM =
(Me.DataBersihTableAdapter.getItemProsesMasuk.Rows.Count())
    'countC1ItemPM = jumlah item c1 dalam proses masuk

    'menghitung jumlah baris masing2 item C1 dalam proses masuk
    'mengambil nama masing2 item C1 dalam proses masuk
    Dim countC1ItemPM(countC1PM) As Integer
    'countC1ItemPM = jumlah row masing2 item C1 dalam PM
    Dim strC1ItemPM(countC1PM) As String
    'strC1ItemPM = nama masing-masing item C1 dalam PM
    For i As Integer = 1 To countC1PM
        strC1ItemPM(i) =
Me.DataBersihTableAdapter.getItemProsesMasuk(i - 1).Item("prosesMasuk")
        countC1ItemPM(i) =
Me.DataBersihTableAdapter.countProsesMasuk(strC1ItemPM(i))
    Next i
    'Didapat nama masing2 item dalam C1 disimpan di array
    strC1ItemPM(countC1PM)
    'Didapat jumlah baris masing2 item dalam C1 disimpan di array
    countC1Ite(countC1PM)

    'MEMBUAT L1
    Dim maxL1PM As Integer = 0
    For i As Integer = 1 To countC1PM
        If countC1ItemPM(i) > tres Then
            maxL1PM = maxL1PM + 1
        End If
    Next i
    'maxL1PM = jumlah item L1 dalam Proses masuk

    Dim countL1ItemPM(maxL1PM) As Integer
    'countL1ItemPM = jumlah baris masing2 item L1 dalam proses
masuk
    Dim strL1ItemPM(maxL1PM) As String
    'strL1ItemPM = nama masing2 item L1 dalam proses masuk
    Dim indexL1PM As Integer = 1
    'indexL1Kat = index baru array L1 dalam proses masuk
    For i As Integer = 1 To countC1PM
        If countC1ItemPM(i) > tres Then
            countL1ItemPM(indexL1PM) = countC1ItemPM(i)
            strL1ItemPM(indexL1PM) = strC1ItemPM(i)
            indexL1PM = indexL1PM + 1
        End If
    Next i

```



```

        'Didapat nama masing2 item dalam L1 disimpan di array
strL1ItemKat(maxL1kat)
        'Didapat jumlah baris masing2 item dalam L1 disimpan di array
countL1ItemKat(maxL1kat)

```

```

'MEMBUAT C2
Dim maxIndexC2KatPM As Integer
maxIndexC2KatPM = maxL1Kat * maxL1PM
'maxindexC2 = jumlah item dalam C2
Dim countC2ItemKatPM(maxIndexC2KatPM) As Integer
'countC2KatPM = jumlah baris masing2 item c2
Dim strC2ItemKatPM(maxIndexC2KatPM) As String
'strC2ItemKatPM = nama gabungan item dalam c2
Dim indexC2KatPM As Integer = 1
For i As Integer = 1 To maxL1Kat
    For j As Integer = 1 To maxL1PM
        countC2ItemKatPM(indexC2KatPM) =
Me.DataBersihTableAdapter.countKatPM(strL1ItemKat(i), strL1ItemPM(j))
        strC2ItemKatPM(indexC2KatPM) = (strL1ItemKat(i) & ","
& strL1ItemPM(j))
        indexC2KatPM = indexC2KatPM + 1
    Next j
Next i
'didapat nama-nama masing item C2 disimpan dalam
strC2ItemKatPM(maxIndexC2)
'didapat jumlah baris masing2 item C2 disimpan dalam
countC2ItemKatPM(maxIndexC2)

```

```

'MEMBUAT L2
Dim maxIndexL2KatPM As Integer
For i As Integer = 1 To maxIndexC2KatPM
    If countC2ItemKatPM(i) > tres Then
        maxIndexL2KatPM = maxIndexL2KatPM + 1
    End If
Next i

Dim CountL2ItemKatPM(maxIndexL2KatPM) As Integer
Dim strL2ItemKatPM(maxIndexL2KatPM) As String
Dim indexL2KatPM As Integer
For i As Integer = 1 To maxIndexC2KatPM
    If countC2ItemKatPM(i) > tres Then
        CountL2ItemKatPM(indexL2KatPM) = countC2ItemKatPM(i)
        strL2ItemKatPM(indexL2KatPM) = strC2ItemKatPM(i)
        sup = CountL2ItemKatPM(indexL2KatPM) /
DataSetGabungan.dataBersih.Rows.Count
        antecedent =
Me.DataBersihTableAdapter.countKategori(Microsoft.VisualBasic.Left(strL2I
temKatPM(indexL2KatPM), 2))
        conf = CountL2ItemKatPM(indexL2KatPM) / antecedent
        conf = conf * 100
        sup = sup * 100
        sup = FormatNumber(sup, 4)
        conf = FormatNumber(conf, 4)

frmreport.ItemSetTableAdapter.InsertQuery(strL2ItemKatPM(indexL2KatPM),
CountL2ItemKatPM(indexL2KatPM), sup, conf)
        indexL2KatPM = indexL2KatPM + 1
    End If

```

```

Next i

ElseIf pil = "Asal Sekolah" Then
    'MEMBUAT C1 ASAL SEKOLAH
    'menghitung jumlah item c1 dalam asal Sekolah
    Dim countClAS As Integer
    countClAS =
(Me.DataBersihTableAdapter.getItemAsalSekolah.Rows.Count())
    'countClItemAS = jumlah item c1 dalam Asal Sekolah
    'menghitung jumlah baris masing2 item C1 dalam Asal Sekolah
    'mengambil nama masing2 item C1 dalam Asal Sekolah
    Dim countClItemAS(countClAS) As Integer
    'countClItemAS = jumlah row masing2 item C1 dalam Asal
Sekolah
    Dim strClItemAS(countClAS) As String
    'strClItemAS = nama masing-masing item C1 dalam Asal Sekolah
    For i As Integer = 1 To countClAS
        strClItemAS(i) =
Me.DataBersihTableAdapter.getItemAsalSekolah(i - 1).Item("asalSekolah")
        countClItemAS(i) =
Me.DataBersihTableAdapter.countAsalSekolah(strClItemAS(i))
    Next i
    'Didapat nama masing2 item dalam C1 disimpan di array
strClItemAS(countClAS)
    'Didapat jumlah baris masing2 item dalam C1 disimpan di array
countClIte(countClAS)

    'MEMBUAT L1
    Dim maxL1AS As Integer = 0
    For i As Integer = 1 To countClAS
        If countClItemAS(i) > tres Then
            maxL1AS = maxL1AS + 1
        End If
    Next i
    'maxL1AS = jumlah item L1 dalam Asal Sekolah

    Dim countL1ItemAS(maxL1AS) As Integer
    'countL1ItemAS = jumlah baris masing2 item L1 dalam Asal
SEkolah
    Dim strL1ItemAS(maxL1AS) As String
    'strL1ItemAS = nama masing2 item L1 dalam Asal Sekolah
    Dim indexL1AS As Integer = 1
    'indexL1Kat = index baru array L1 dalam Asal Sekolah
    For i As Integer = 1 To countClAS
        If countClItemAS(i) > tres Then
            countL1ItemAS(indexL1AS) = countClItemAS(i)
            strL1ItemAS(indexL1AS) = strClItemAS(i)
            indexL1AS = indexL1AS + 1
        End If
    Next i
    'Didapat nama masing2 item dalam L1 disimpan di array
strL1ItemKat(maxL1kat)
    'Didapat jumlah baris masing2 item dalam L1 disimpan di array
countL1ItemKat(maxL1kat)

    'MEMBUAT C2
    Dim maxIndexC2KatAS As Integer

```

```

maxIndexC2KatAS = maxL1Kat * maxL1AS
'maxindexC2 = jumlah item dalam C2
Dim countC2ItemKatAS(maxIndexC2KatAS) As Integer
'countC2KatAS = jumlah baris masing2 item c2
Dim strC2ItemKatAS(maxIndexC2KatAS) As String
'strC2ItemKatAS = nama gabungan item dalam c2
Dim indexC2KatAS As Integer = 1
For i As Integer = 1 To maxL1Kat
    For j As Integer = 1 To maxL1AS
        countC2ItemKatAS(indexC2KatAS) =
Me.DataBersihTableAdapter.countKatAsalSekolah(strL1ItemKat(i),
strL1ItemAS(j))
        strC2ItemKatAS(indexC2KatAS) = (strL1ItemKat(i) & ","
& strL1ItemAS(j))
        indexC2KatAS = indexC2KatAS + 1
    Next j
Next i
'didapat nama-nama masing item C2 disimpan dalam
strC2ItemKatAS(maxIndexC2)
'didapat jumlah baris masing2 item C2 disimpan dalam
countC2ItemKatAS(maxIndexC2)

'MEMBUAT L2
Dim maxIndexL2KatAS As Integer
For i As Integer = 1 To maxIndexC2KatAS
    If countC2ItemKatAS(i) > tres Then
        maxIndexL2KatAS = maxIndexL2KatAS + 1
    End If
Next i

Dim CountL2ItemKatAS(maxIndexL2KatAS) As Integer
Dim strL2ItemKatAS(maxIndexL2KatAS) As String
Dim indexL2KatAS As Integer
For i As Integer = 1 To maxIndexC2KatAS
    If countC2ItemKatAS(i) > tres Then
        CountL2ItemKatAS(indexL2KatAS) = countC2ItemKatAS(i)
        strL2ItemKatAS(indexL2KatAS) = strC2ItemKatAS(i)
        sup = CountL2ItemKatAS(indexL2KatAS) /
DataSetGabungan.dataBersih.Rows.Count
        antecedent =
Me.DataBersihTableAdapter.countKategori(Microsoft.VisualBasic.Left(strL2I
temKatAS(indexL2KatAS), 2))
        conf = CountL2ItemKatAS(indexL2KatAS) / antecedent
        conf = conf * 100
        sup = sup * 100
        sup = FormatNumber(sup, 4)
        conf = FormatNumber(conf, 4)

frmreport.ItemSetTableAdapter.InsertQuery(strL2ItemKatAS(indexL2KatAS),
CountL2ItemKatAS(indexL2KatAS), sup, conf)
        indexL2KatAS = indexL2KatAS + 1
    End If
Next i

ElseIf pil = "Kota Asal" Then
'MEMBUAT C1 KOTA ASAL
'menghitung jumlah item c1 dalam Kota Asal
Dim countC1KotaAsal As Integer

```

```

        countC1KotaAsal =
(Me.DataBersihTableAdapter.GetItemAsalKota.Rows.Count())

        'menghitung jumlah baris masing2 item C1 dalam Jurusan
        'mengambil nama masing2 item C1 dalam Jurusan
        Dim countC1ItemKotaAsal(countC1KotaAsal) As Integer
        'countC1ItemPM = jumlah row masing2 item C1 dalam Program
Studi
        Dim strC1ItemKotaAsal(countC1KotaAsal) As String
        'strC1ItemPM = nama masing-masing item C1 dalam Program Studi
        For i As Integer = 1 To countC1KotaAsal
            strC1ItemKotaAsal(i) =
Me.DataBersihTableAdapter.GetItemAsalKota(i - 1).Item("kotaSekolah")
            countC1ItemKotaAsal(i) =
Me.DataBersihTableAdapter.countAsalKota(strC1ItemKotaAsal(i))
        Next i
        'Didapat nama masing2 item dalam C1 disimpan di array
strC1ItemKotaAsal(countC1KotaAsal)
        'Didapat jumlah baris masing2 item dalam C1 disimpan di array
countC1ItemKotaAsal(countC1KotaAsal)

        'MEMBUAT L1
        Dim maxL1KotaAsal As Integer = 0
        For i As Integer = 1 To countC1KotaAsal
            If countC1ItemKotaAsal(i) > tres Then
                maxL1KotaAsal = maxL1KotaAsal + 1
            End If
        Next i
        'maxL1KotaAsal = jumlah item L1 dalam Program Studi
        Dim countL1ItemKotaAsal(maxL1KotaAsal) As Integer
        'countL1ItemKotaAsal = jumlah baris masing2 item L1 dalam
Program Studi
        Dim strL1ItemKotaAsal(maxL1KotaAsal) As String
        'strL1ItemKotaAsal = nama masing2 item L1 dalam Program Studi
        Dim indexL1KotaAsal As Integer = 1
        'indexL1KotaAsal = index baru array L1 dalam Program Studi
        For i As Integer = 1 To countC1KotaAsal
            If countC1ItemKotaAsal(i) > tres Then
                countL1ItemKotaAsal(indexL1KotaAsal) =
countC1ItemKotaAsal(i)
                strL1ItemKotaAsal(indexL1KotaAsal) =
strC1ItemKotaAsal(i)
                indexL1KotaAsal = indexL1KotaAsal + 1
            End If
        Next i

        'MEMBUAT C2
        Dim maxIndexC2KatKotaAsal As Integer
        maxIndexC2KatKotaAsal = maxL1Kat * maxL1KotaAsal
        'maxindexC2 = jumlah item dalam C2
        Dim countC2ItemKatKotaAsal(maxIndexC2KatKotaAsal) As Integer
        'countC2KatPM = jumlah baris masing2 item c2
        Dim strC2ItemKatKotaAsal(maxIndexC2KatKotaAsal) As String
        'strC2ItemKatPM = nama gabungan item dalam c2
        Dim indexC2KatKotaAsal As Integer = 1
        For i As Integer = 1 To maxL1Kat
            For j As Integer = 1 To maxL1KotaAsal
                countC2ItemKatKotaAsal(indexC2KatKotaAsal) =
Me.DataBersihTableAdapter.countKatKota(strL1ItemKat(i),
strL1ItemKotaAsal(j))
            
```

```

        strC2ItemKatKotaAsal(indexC2KatKotaAsal) =
(strL1ItemKat(i) & "," & strL1ItemKotaAsal(j))
        indexC2KatKotaAsal = indexC2KatKotaAsal + 1
    Next j
Next i

'MEMBUAT L2
Dim maxIndexL2KatKotaAsal As Integer
For i As Integer = 1 To maxIndexC2KatKotaAsal
    If countC2ItemKatKotaAsal(i) > tres Then
        maxIndexL2KatKotaAsal = maxIndexL2KatKotaAsal + 1
    End If
Next i
Dim countL2ItemKatKotaAsal(maxIndexL2KatKotaAsal) As Integer
Dim strL2ItemKatKotaAsal(maxIndexL2KatKotaAsal) As String
Dim indexL2KatKotaAsal As Integer
For i As Integer = 1 To maxIndexC2KatKotaAsal
    If countC2ItemKatKotaAsal(i) > tres Then
        countL2ItemKatKotaAsal(indexL2KatKotaAsal) =
countC2ItemKatKotaAsal(i)
        strL2ItemKatKotaAsal(indexL2KatKotaAsal) =
strC2ItemKatKotaAsal(i)
        sup = countL2ItemKatKotaAsal(indexL2KatKotaAsal) /
DataSetGabungan.dataBersih.Rows.Count
        antecedent =
Me.DataBersihTableAdapter.countKategori(Microsoft.VisualBasic.Left(strL2I
temKatKotaAsal(indexL2KatKotaAsal), 2))
        conf = countL2ItemKatKotaAsal(indexL2KatKotaAsal) /
antecedent

        conf = conf * 100
        sup = sup * 100
        sup = FormatNumber(sup, 4)
        conf = FormatNumber(conf, 4)

frmreport.ItemSetTableAdapter.InsertQuery(strL2ItemKatKotaAsal(indexL2Kat
KotaAsal), countL2ItemKatKotaAsal(indexL2KatKotaAsal), sup, conf)
        indexL2KatKotaAsal = indexL2KatKotaAsal + 1
    End If
Next i

ElseIf pil = "Jurusan" Then

'MEMBUAT C1 Jurusan
'menghitung jumlah item c1 dalam Jurusan
Dim countC1Prodi As Integer
countC1Prodi =
(Me.DataBersihTableAdapter.getItemProdi.Rows.Count())

'menghitung jumlah baris masing2 item C1 dalam Jurusan
'mengambil nama masing2 item C1 dalam Jurusan
Dim countC1ItemProdi(countC1Prodi) As Integer
'countC1ItemPM = jumlah row masing2 item C1 dalam Jurusan
Dim strC1ItemProdi(countC1Prodi) As String
'strC1ItemPM = nama masing-masing item C1 dalam Jurusan
For i As Integer = 1 To countC1Prodi
    strC1ItemProdi(i) =
Me.DataBersihTableAdapter.getItemProdi(i - 1).Item("programStudi")
    countC1ItemProdi(i) =
Me.DataBersihTableAdapter.countProdi(strC1ItemProdi(i))

```

```

        Next i
        'Didapat nama masing2 item dalam C1 disimpan di array
strC1ItemProdi(countC1Prodi)
        'Didapat jumlah baris masing2 item dalam C1 disimpan di array
countC1ItemProdi(countC1Prodi)

'MEMBUAT L1
Dim maxL1Prodi As Integer = 0
For i As Integer = 1 To countC1Prodi
    If countC1ItemProdi(i) > tres Then
        maxL1Prodi = maxL1Prodi + 1
    End If
Next i
'maxL1Prodi = jumlah item L1 dalam Jurusan
Dim countL1ItemProdi(maxL1Prodi) As Integer
'countL1ItemProdi = jumlah baris masing2 item L1 dalam
Jurusan
Dim strL1ItemProdi(maxL1Prodi) As String
'strL1ItemProdi = nama masing2 item L1 dalam Jurusan
Dim indexL1Prodi As Integer = 1
'indexL1Prodi = index baru array L1 dalam Jurusan
For i As Integer = 1 To countC1Prodi
    If countC1ItemProdi(i) > tres Then
        countL1ItemProdi(indexL1Prodi) = countC1ItemProdi(i)
        strL1ItemProdi(indexL1Prodi) = strC1ItemProdi(i)
        indexL1Prodi = indexL1Prodi + 1
    End If
Next i

'MEMBUAT C2
Dim maxIndexC2KatProdi As Integer
maxIndexC2KatProdi = maxL1Kat * maxL1Prodi
'maxindexC2 = jumlah item dalam C2
Dim countC2ItemKatProdi(maxIndexC2KatProdi) As Integer
'countC2KatPM = jumlah baris masing2 item c2
Dim strC2ItemKatProdi(maxIndexC2KatProdi) As String
'strC2ItemKatPM = nama gabungan item dalam c2
Dim indexC2KatProdi As Integer = 1
For i As Integer = 1 To maxL1Kat
    For j As Integer = 1 To maxL1Prodi
        countC2ItemKatProdi(indexC2KatProdi) =
Me.DataBersihTableAdapter.countKatProdi(strL1ItemKat(i),
strL1ItemProdi(j))
        strC2ItemKatProdi(indexC2KatProdi) = (strL1ItemKat(i)
& "," & strL1ItemProdi(j))
        indexC2KatProdi = indexC2KatProdi + 1
    Next j
Next i

'MEMBUAT L2
Dim maxIndexL2KatProdi As Integer
For i As Integer = 1 To maxIndexC2KatProdi
    If countC2ItemKatProdi(i) > tres Then
        maxIndexL2KatProdi = maxIndexL2KatProdi + 1
    End If
Next i
Dim CountL2ItemKatProdi(maxIndexL2KatProdi) As Integer
Dim strL2ItemKatProdi(maxIndexL2KatProdi) As String
Dim indexL2KatProdi As Integer
For i As Integer = 1 To maxIndexC2KatProdi

```

```

        If countC2ItemKatProdi(i) > tres Then
            CountL2ItemKatProdi(indexL2KatProdi) =
countC2ItemKatProdi(i)
            strL2ItemKatProdi(indexL2KatProdi) =
strC2ItemKatProdi(i)
            sup = CountL2ItemKatProdi(indexL2KatProdi) /
DataSetGabungan.dataBersih.Rows.Count
            antecedent =
Me.DataBersihTableAdapter.countKategori(Microsoft.VisualBasic.Left(strL2I
temKatProdi(indexL2KatProdi), 2))
            conf = CountL2ItemKatProdi(indexL2KatProdi) /
antecedent

            conf = conf * 100
            sup = sup * 100
            sup = FormatNumber(sup, 4)
            conf = FormatNumber(conf, 4)

frmreport.ItemSetTableAdapter.InsertQuery(strL2ItemKatProdi(indexL2KatPro
di), CountL2ItemKatProdi(indexL2KatProdi), sup, conf)
            indexL2KatProdi = indexL2KatProdi + 1
        End If
    Next i
End If
Me.Visible = False
frmreport.Show()
End Sub

Private Sub GabunganBindingNavigatorSaveItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
    Me.Validate()
    Me.TableAdapterManager.UpdateAll(Me.DataSetGabungan)
End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
    End
End Sub

Private Sub CmbBoxPilJur_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles CmbBoxPilJur.SelectedIndexChanged
    tolbar.Text = "Jurusan yang di pilih adalah = " +
CmbBoxPilJur.SelectedItem + ", Pilih Atribut yang akan di proses "
    lblJur.Text = "Program studi yang dipilih adalah " +
CmbBoxPilJur.SelectedItem
    cmbBoxDm.Enabled = True
    cmbBoxDm.Focus()
    If CmbBoxPilJur.SelectedItem <> "Semua Jurusan" Then
        cmbBoxDm.Items.Remove("Jurusan")
    End If
    If CmbBoxPilJur.SelectedItem = "Semua Jurusan" Then
        cmbBoxDm.Items.Add("Jurusan")
    End If
    If cmbBoxDm.Items.Count >= 5 Then
        cmbBoxDm.Items.Remove("Jurusan")
    End If

End Sub

Private Sub domtres_ValueChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles domtres.ValueChanged
    btnProses.Enabled = True
    btnProses.Focus()
    tolbar.Text = "nilai threshold = " + domtres.Value.ToString + ",
klik tombol proses untuk memulai proses"

```

```

        End Sub
        Private Sub btnProses_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnProses.GotFocus
            If domtres.Value > jmlRecordDataBersih Then
                domtres.Focus()
                MsgBox("threshold tidak boleh melebihi jumlah record, jumlah
record = " & jmlRecordDataBersih)
            End If
        End Sub
    End Sub
End Class

'Hal report
Public Class FormReport
    Private Sub BtnKeluar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnKeluar.Click
        End
    End Sub
    Private Sub BtnKembali_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnKembali.Click
        formAwal.Show()
        formAwal.btnProses.Enabled = True
        Me.Dispose()
    End Sub
    Private Sub ItemSetBindingNavigatorSaveItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        Me.Validate()
        Me.ItemSetBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.DataSetGabungan)
    End Sub
    Private Sub FormReport_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        tolbar2.Text = "Hasil Proses Data mining, dengan jurusan = " +
formAwal.CmbBoxPilJur.SelectedItem + ", item = " +
formAwal.cmbBoxDm.SelectedItem
        LblNilJur.Text = Dm.formAwal.CmbBoxPilJur.SelectedItem
        If LblNilJur.Text = "" Then
            LblNilJur.Text = "Semua Jurusan"
        End If
        'bikin report yang sebelah kiri
        'TODO: This line of code loads data into the
'DataSetGabungan.itemSet' table. You can move, or remove it, as needed.
        Me.ItemSetTableAdapter.Fill(Me.DataSetGabungan.itemSet)
        'bikin report kiri
        Dim countKat(6) As Integer
        Dim nilKatConf(6) As Double
        Dim nilKatConfS(6) As Double
        Dim nilKatSupp(6) As Double
        Dim itemKatConf(6) As String
        Dim itemKatSupp(6) As String
        BtnKeluar.Focus()
        'menghitung masingmasing kategori
        For i As Integer = 1 To 3
            countKat(i) = Me.ItemSetTableAdapter.count("A" & i & "%")
            If countKat(i) > 0 Then
                nilKatConf(i) = Me.ItemSetTableAdapter.GetNil("A" & i &
"%") (0).Item("confidence")
                nilKatSupp(i) = Me.ItemSetTableAdapter.getNilSupp("A" & i
& "%") (0).Item("Support")
                itemKatConf(i) = Me.ItemSetTableAdapter.GetNil("A" & i &
"%") (0).Item("Item Set")
            End If
        Next i
    End Sub
End Class

```



```

        itemKatSupp(i) = Me.ItemSetTableAdapter.getNilSupp("A" &
i & "%")(0).Item("Item Set")
        nilKatConfS(i) = Me.ItemSetTableAdapter.GetNil("A" & i &
"%")(0).Item("Support")
    Else

        nilKatConf(i) = 0
        nilKatSupp(i) = 0
        itemKatConf(i) = 0
        itemKatSupp(i) = 0
        nilKatConfS(i) = 0
    End If

Next i
For i As Integer = 1 To 3
    countKat(3 + i) = Me.ItemSetTableAdapter.count("B" & i & "%")
    If countKat(3 + i) > 0 Then
        nilKatConf(3 + i) = Me.ItemSetTableAdapter.GetNil("B" & i
& "%")(0).Item("confidence")
        nilKatSupp(3 + i) = Me.ItemSetTableAdapter.getNilSupp("B"
& i & "%")(0).Item("Support")
        itemKatConf(3 + i) = Me.ItemSetTableAdapter.GetNil("B" &
i & "%")(0).Item("Item Set")
        itemKatSupp(3 + i) =
Me.ItemSetTableAdapter.getNilSupp("B" & i & "%")(0).Item("Item Set")
        nilKatConfS(3 + i) = Me.ItemSetTableAdapter.GetNil("B" &
i & "%")(0).Item("Support")

    Else
        nilKatConf(3 + i) = 0
        nilKatSupp(3 + i) = 0
        itemKatConf(3 + i) = 0
        itemKatSupp(3 + i) = 0
        nilKatConfS(3 + i) = 0
    End If
Next i
If countKat(1) = 0 Then
    lblRpt1.Text = "data kosong atau tidak mencapai minimum
thrershold, " & ControlChars.NewLine & "informasi tidak dapat
ditampilkan."
Else

    lblRpt1.Text = "Memiliki Hubungan dengan " & itemKatConf(1) &
" dengan nilai kepercayaan = " & nilKatConf(1) & "%" &
ControlChars.NewLine & "Hal ini didukung oleh " & nilKatConfS(1) & "%
dari seluruh data merupakan kombinasi keduanya."
    End If
    If countKat(2) = 0 Then

        lblRpt2.Text = "data kosong atau tidak mencapai minimum
thrershold, " & ControlChars.NewLine & "informasi tidak dapat
ditampilkan."
    Else

        lblRpt2.Text = "Memiliki Hubungan dengan " & itemKatConf(2) &
" dengan nilai kepercayaan = " & nilKatConf(2) & "%" &
ControlChars.NewLine & "Hal ini didukung oleh " & nilKatConfS(2) & "%
dari seluruh data merupakan kombinasi keduanya."
    End If

```

```

        If countKat(3) = 0 Then

            LblRpt3.Text = "data kosong atau tidak mencapai minimum
thrershold, " & ControlChars.NewLine & "informasi tidak dapat
ditampilkan."
        Else
            LblRpt3.Text = "Memiliki Hubungan dengan " & itemKatConf(3) &
" dengan nilai kepercayaan = " & nilKatConf(3) & "%" &
ControlChars.NewLine & "Hal ini didukung oleh " & nilKatConfS(3) & "%
dari seluruh data merupakan kombinasi keduanya."
        End If
        If countKat(4) = 0 Then
            LblRpt4.Text = "data kosong atau tidak mencapai minimum
thrershold, " & ControlChars.NewLine & "informasi tidak dapat
ditampilkan."
        Else
            LblRpt4.Text = "Memiliki Hubungan dengan " & itemKatConf(4) &
" dengan nilai kepercayaan = " & nilKatConf(4) & "%" &
ControlChars.NewLine & "Hal ini didukung oleh " & nilKatConfS(4) & "%
dari seluruh data merupakan kombinasi keduanya."
        End If

        If countKat(5) = 0 Then

            lblRpt5.Text = "data kosong atau tidak mencapai minimum
thrershold, " & ControlChars.NewLine & "informasi tidak dapat
ditampilkan."
        Else
            lblRpt5.Text = "Memiliki Hubungan dengan " & itemKatConf(5) &
" dengan nilai kepercayaan = " & nilKatConf(5) & "%" &
ControlChars.NewLine & "Hal ini didukung oleh " & nilKatConfS(5) & "%
dari seluruh data merupakan kombinasi keduanya."
        End If

        If countKat(6) = 0 Then
            lblRpt6.Text = "data kosong atau tidak mencapai minimum
thrershold, " & ControlChars.NewLine & "informasi tidak dapat
ditampilkan."
        Else

            lblRpt6.Text = "Memiliki Hubungan dengan " & itemKatConf(6) &
" dengan nilai kepercayaan = " & nilKatConf(6) & "%" &
ControlChars.NewLine & "Hal ini didukung oleh " & nilKatConfS(6) & "%
dari seluruh data merupakan kombinasi keduanya."
        End If
        'membuat report teks
    End Sub
End Class

```